Navy Atmospheric Boundary Layer (NABL) Model System: Software/Interface Design Document

P. A. Petit
ST Systems Corporation
P. O. Box 51306
Pacific Grove, CA 93950

Prepared for:
Atmospheric Directorate
Monterey, CA 93943-5006
Contract No. N00014-88-D-0333

ABSTRACT

This document describes the complete design of the Naval Atmospheric Boundary Layer (NABL) Model System in terms of its Computer Software Components (CSCs) and Computer Software Units (CSUs).

ACKNOWLEDGEMENTS

The support of the sponsor, Fleet Numerical Oceanography Center, Mr. Leo C. Clarke, Program Element O&M,N, is gratefully acknowledged. The Naval Oceanographic and Atmospheric Research Laboratory, Atmospheric Directorate project coordinator for this effort was Mr. Sam Brand.

CONTENTS

CONTENTS (cont'd)

CONTENTS (cont'd)

FIGURES

APPENDIXES

NAVY ATMOSPHERIC BOUNDARY LAYER (NABL) MODEL SYSTEM: SOFTWARE/INTERFACE DESIGN DOCUMENT

1 SCOPE.

1.1 Identification. This design document applies to the Naval Atmospheric Boundary Layer (NABL) Model System which is identified and abbreviated as NABLS. Its identification number is "to be determined". This design is derived from the NABLS requirements document (reference a, in section 2. below).

1.2 System overview. The NABLS will implement the "NABL model" on the Fleet Numerical Oceanography Center (FNOC), Monterey, CA computer system. The NABL model was developed by the Naval Oceanographic and Atmospheric Research Laboratory (NOARL), Atmospheric Directorate, Monterey CA to produce short-range (0 to 36 hour) forecasts of meteorological conditions in the atmospheric boundary layer. Although NABL is a one-dimensional "stick" model, it uses turbulence parametrizations and a vertical resolution which are superior to those in the current Navy operational regional and global atmospheric prediction model systems (NORAPS and NOGAPS respectively). NABL obtains adiabatic tendencies of temperature and moisture from NORAPS or NOGAPS, giving it three-dimensional character, although NABL does not interact in a two-way sense with either of the larger-scale prediction model systems. NABL has demonstrated superior skill at predicting the vertical structure of temperature and refractivity in the boundary layer (refs. g and h.).

The NABLS will permit a system user to specify where geographically and for what forecast intervals and times the NABL model will be run; will cause the requisite initial condition and tendency information from one of the larger-scale models to be made available for input to NABLS; and will provide for saving the resultant, high-resolution NABL model output in a convenient format(s) for use by such other applications (external to NABLS) which may now or in the future have need for such information.

1.3 Document overview. This document describes the complete design of the NABLS in terms of its Computer Software Components (CSCs) and Computer Software Units (CSUs). It would normally be used as a basis for coding each CSU but that is not true in this instance for reasons explained in the Note at the end of this subparagraph. This document is organized according to the DOD 2167A standard for the Software Design Document (SDD) (ref. b,c.). That SRS standard has been modified slightly to incorporate into that one document the pertinent, non-redundant sections of the DOD 2167A standard for the Interface Design Document (IDD) (ref. d.). This modification caused section 3 of the SDD outline to be expanded to include an additional subsection 3.3 entitled Interface Description.

Section 1 identifies and provides an overview of the NABLS. Section 2 lists applicable documents. Section 3 provides an overview of the NABLS, describes its various components, and provides a preliminary design in terms of execution control and data flow between those components. Section 4 provided design

details to the unit level. Section 5 describes the global data elements within the NABLS. Section 6 describes the data files which are shared between NABLS components. Section 7 provides traceability of the requirements stated in the Requirements Specification (ref. a) to the paragraphs in this document.

Note: At the time this document was being prepared, the NABL model and various interfaces thereto had already been installed at FNOC and the "system" as such was undergoing informal operational checkout. This design therefore documents the requirements for and architecture of the NABLS retrospectively. Accordingly, emphasis herein is given to providing information which will benefit the maintenance programmer for the system as it now exists and/or the analyst who may wish to make improvements to NABLS as it is now implemented. Requirements and design specifications in this document which vary substantially from NABLS as it is now implemented are noted (in bold face) as potential changes or enhancements which could be made at a later date.

2. REFERENCED DOCUMENTS.

2.1 Government documents. The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

a. Navy Atmospheric Boundary Layer (NABL) Model System: Software/Interface Requirements Specification (NABLS SRS/IRS). NOARL Technical Note 182, November 1991.

b. DOD-STD-2167A, Military Standard, Defense System Software Development, DoD, Washington D.C., 29 February 1988.

c. DI-MCCR-80012A, Data Item Description for SDD.

d. DI-MCCR-80027A, Data Item Description for IDD.

e. FNOC Computer User Guide Edition 2 (updated).

f. FNOC Cyber 205 User Guide, December 1989.

g. Vogel, G. N., and R. H. Langland, 1991: An Evaluation of the Navy Atmospheric Boundary Layer (NABL) Model. NOARL Technical Note 160, August 1991.

h. Vogel, G. N., 1991: Assessment and Short-range Forecasting of Tropospheric Ducting: Comparison of Results from the Navy's Local, Regional and Global Forecast Model. NOARL Report 23, 1991.

i. Lewit, H. (CSC), NGPOINT, The NABL NOGAPS Geographic Point File Interface System, User's Instructions, 18 Dec 1990.

j. Lewit, H. (CSC), NRPNABL, The NABL NORAPS Geographic Point File Interface System, User's Instructions, 20 Dec 1990.

k. FLENUMOCEANCENINST 5234.5 (CH-1), FLENUMOCEANCEN Software Standards, 12 January 1990.

Copies of specifications, standards, drawings, and publications required by suppliers in connection with specified procurement functions should be obtained from the contracting agency or as directed by the contracting officer.

2.2 Non-Government documents. The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

l. CDC NOS/BE Reference Manual (available from FNOC/NOARL library).

m. CDC VSOS Version 2 Reference Manual, Volume 1 (available from FNOC/NOARL library).

n. CDC FORTRAN 200 Version 1 Reference Manual (available from FNOC/NOARL library).

3. PRELIMINARY DESIGN. Note: This section would normally be completed and then reviewed as a true "preliminary" cut at system design. The "detailed" design sections which follow would be based on that design review. Because this document is being prepared retrospectively, this section provides a top-level view of the existing system and is in no way preliminary. The section title is retained, however, to comply with the documentation standard (ref. c).

3.1 NABLS overview. Figures 3-1, 3-2 and 3-3 are reproduced herein from ref. a. Figure 3-1 depicts the relationship of the NABLS to its several external interfaces. The figure is also known as the context diagram because it shows the domain of the system. Detailed specifications for each external interface are provided in ref. a.

Figures 3-2 and 3-3 present data flow within the NABLS. The symbology used in these data flow diagrams (DFDs) is described in Appendix A.

3.1.1 NABLS architecture. Figure 3-4 shows the overall NABLS architecture. It relates each requirement (keyed to the identifying numbers assigned to the "bubbles" in figures 3-2 and 3-3) to the programs and procedures which support those requirements. Note: requirement/bubble 1.4 (Transfer Model Output) is not shown because of its loose relationship to the rest of NABLS, as discussed in paragraph 3.2.1.4 of ref. a.

The programs and procedures shown on figure 3-4 constitute the top-level Computer Software Components (CSCs) of the NABLS. They are highlighted by their double-sided boxes. All of these CSCs already exist on the FNOC computer system. Some, as discussed in ref. a and as will be noted again in this document, are recommended for modification and/or consolidation.

Note: In this document the CSCs are discussed as separate entities as they existed on or about 25 October 1991. That caveat is required since, even as this document was being drafted, changes were being made to program NABL to make that primary element of NABLS run more efficiently on both the FED ("front-end") and SAM ("super") computer systems. Some of those recent changes were in direct response to recommendations noted in ref. a.

3.1.2 System states and modes. The NABLS, and in particular program NABL, is designed to operate in the batch mode as part of a scheduled job(s) - either under NOS/BE ((CDC's) Network Operating System/Batch Environment (ref. l)) on FED and/or under VSOS ((CDC's) Virtual Storage Operating System (ref. m)) on SAM. Program NABL satisfies requirements/bubbles 2 through 4 in figure 3-2.

The system may also be operated in an unscheduled batch mode. This would be the usual mode of exercising the INTERFACE LARGER-SCALE MODELS capability (requirement/bubble 1) which requires interactive input from a user; but, only when the user initiates the process. Program NABL may also be exercised in an unscheduled batch mode for R&D applications.

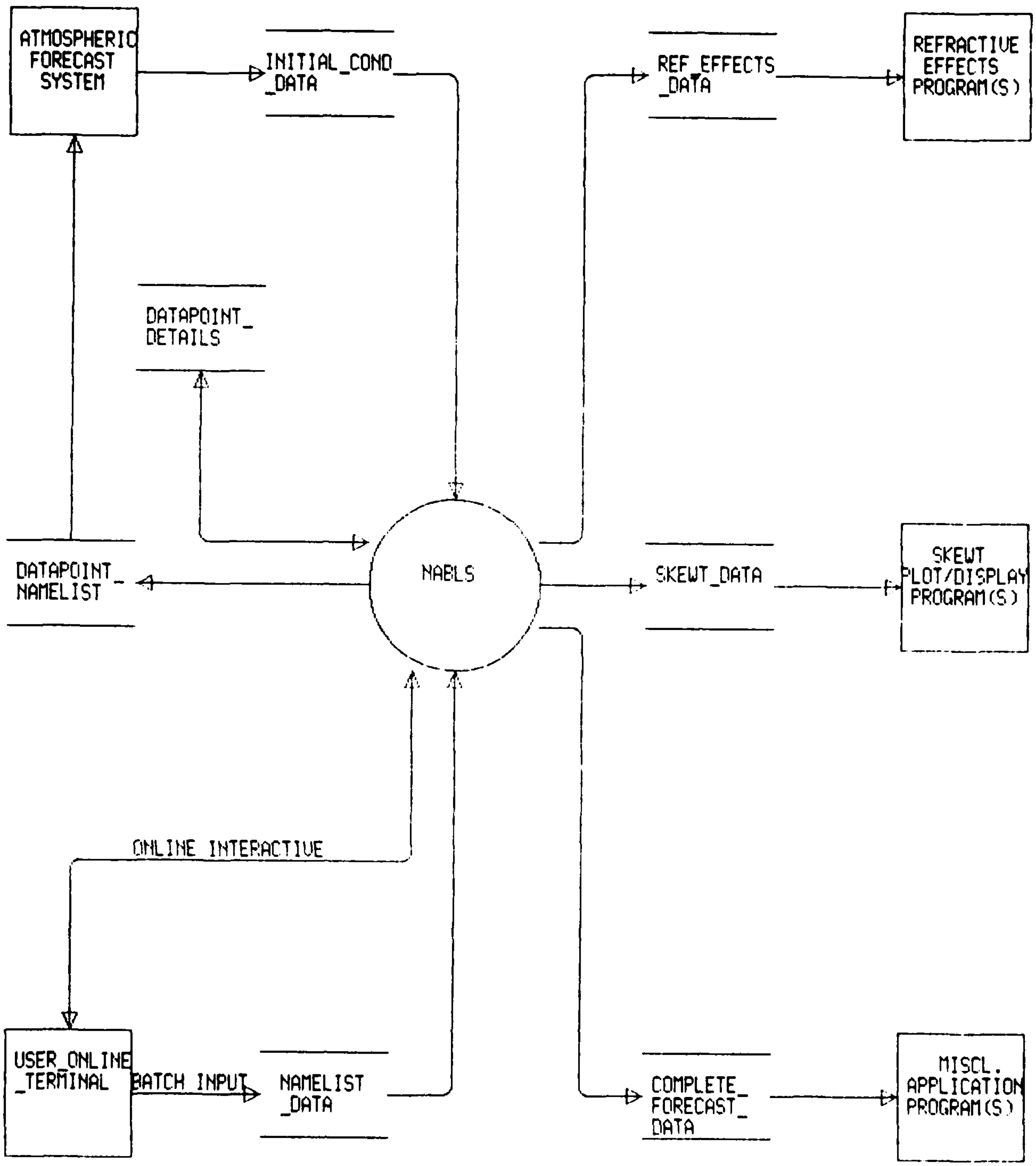

Figure 3-1. Context diagram of NABL System.

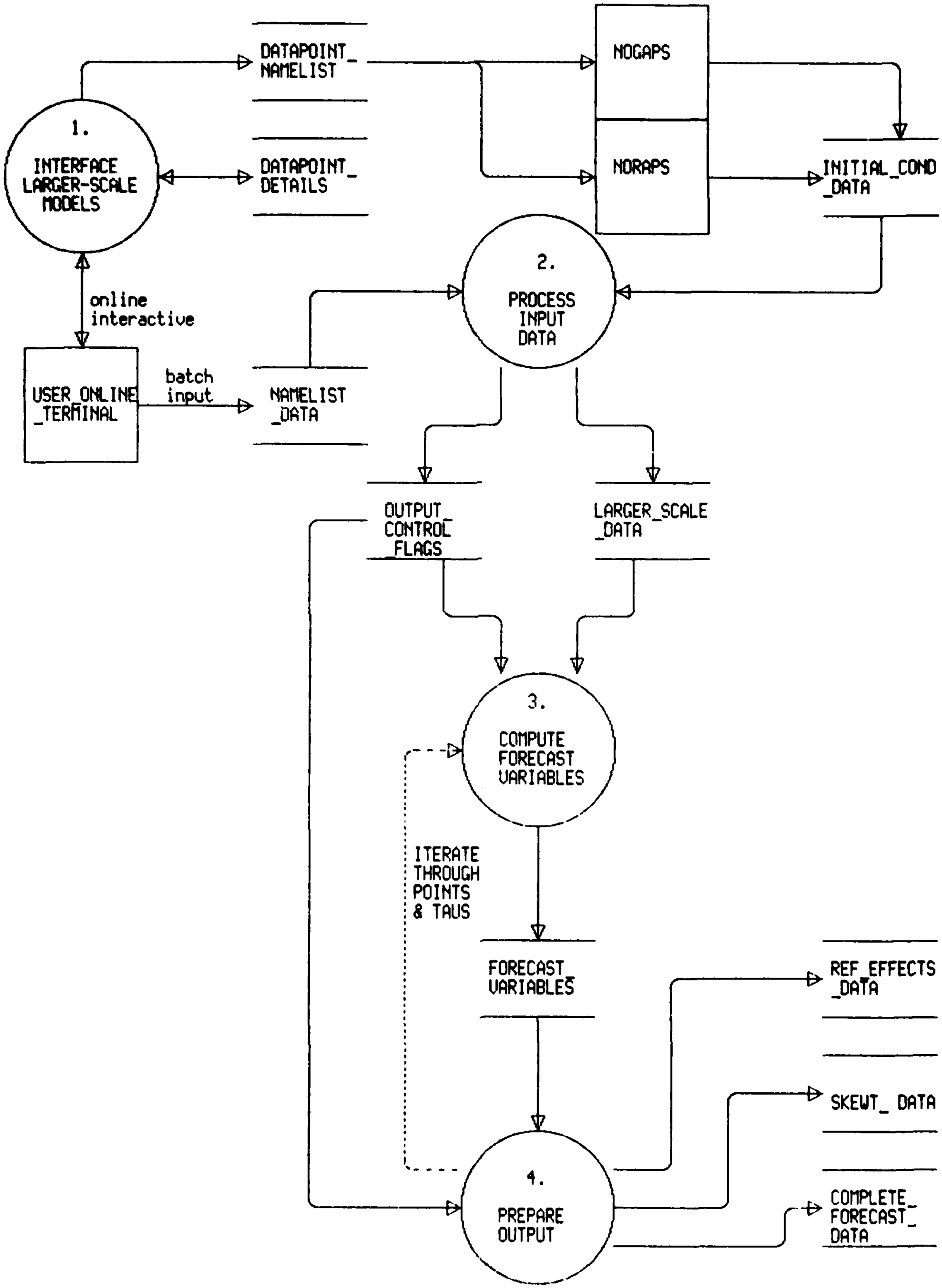

Figure 3-2. Top level DFD of NABL System.

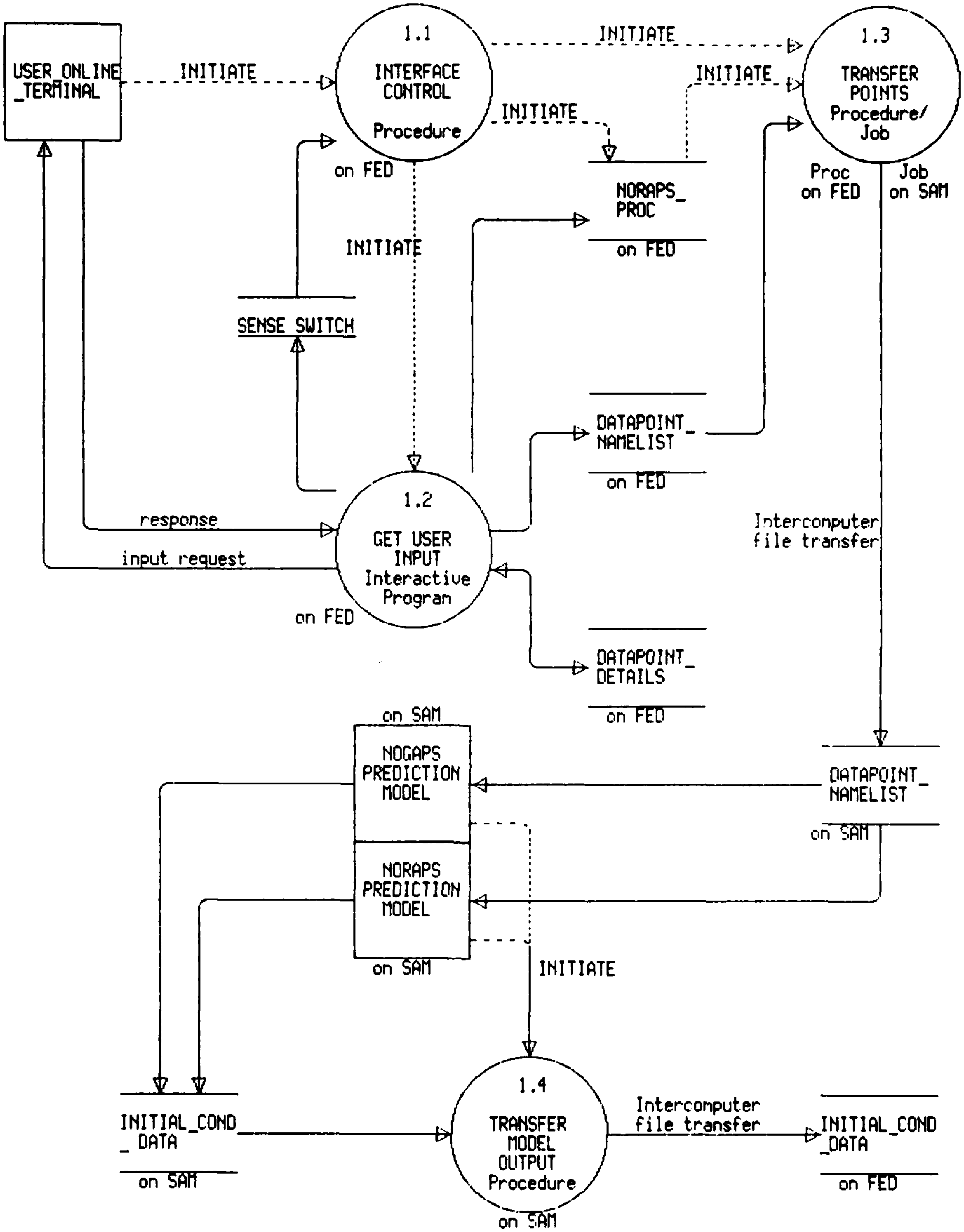

Figure 3-3. DFD of Interface Larger-scale Models (Bubble 1).

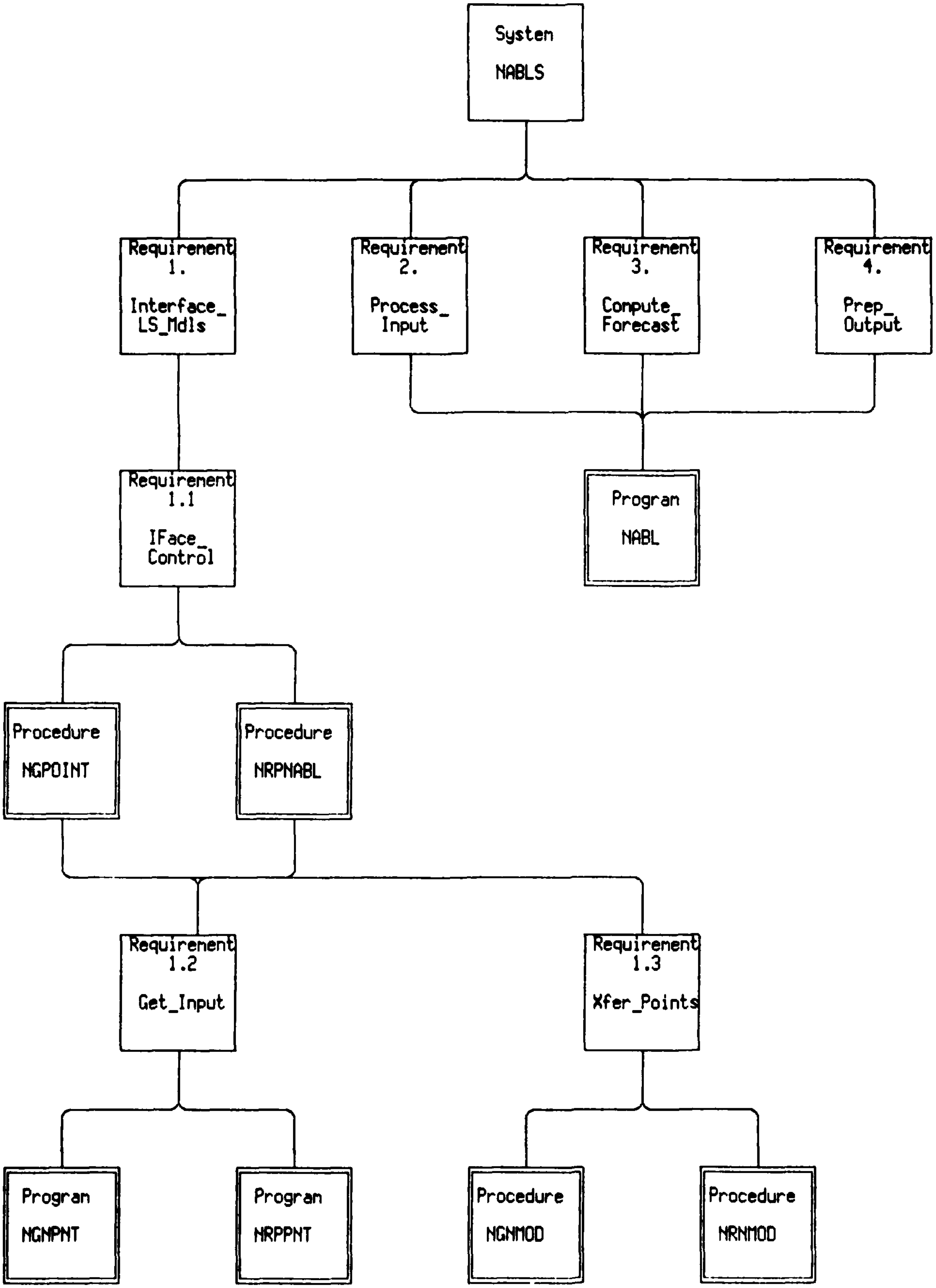

Figure 3-4. The NABLS Architecture.

The NAMELIST_DATA which accompanies the NABL initiating job will also be updated on an unscheduled basis. This will be accomplished by the responsible program maintenance person preparing and submitting an AUTOF update to the AUTO File as described in ref. e (section 7.6).

3.1.3 Memory and processing time allocation. The NABLS CSCs will be governed by the programming standards and resource usage provisions of refs. e (section 6) and f.

3.2 NABLS design description. This section provides a design description of the NABLS CSCs. Each subparagraph below corresponds to one of the seven programs/procedures identified by a double-sided box in figure 3-4.

3.2.1 Program NABL (NABL). This batch-mode atmospheric prediction program is central to the NABLS. This one-dimensional boundary layer model is initialized with profiles of wind, temperature, and moisture at preselected locations as provided from either NORAPS or NOGAPS. The evolution of these profiles is forecast based on parametrized physical processes, surface forcing, and synoptic-scale adiabatic advective tendencies.

Program NABL satisfies the Process Input Data, Compute Forecast Variables, and Prepare Output requirements (paragraphs 3.2.2, 3.2.3 and 3.2.4 of ref. a). A table of the COMPLETE_FORECAST_DATA file data elements may be found in Appendix D.

Control of program execution is provided by the NAMELIST data read in by NABL from the NAMELIST_DATA interface/file (paragraph 3.1.2 of ref. a). A table of the NAMELIST_DATA file data elements may be found in Appendix D.

The individual modules, the Computer Software Units (CSUs), which comprise the NABL program are presented in Section 4.1.

3.2.2 Procedure NGPOINT (NGPOINT). This Cyber Control Language (CCL) procedure is initiated by a user who wishes to input a list or a change to an existing list of NABLS geographic point locations for input to the NOGAPS synoptic scale atmospheric prediction system. This list will control the location of profiles prepared by NOGAPS for input to program NABL.

This procedure, which is described more fully in ref. i, starts the interactive interface program NGNPNT. If warranted, based upon "sense switch" setting provided by that program, this procedure will cause the resultant DATAPOINT_NAMELIST to be transferred into the SAM "super" computer system for use the next time the NOGAPS is executed.

Procedure NGPOINT satisfies the Interface Control requirement (paragraph 3.2.1.1 or ref. a) for NOGAPS.

Control of procedure execution is provided by the sense switch settings mentioned above - only if SW1 is set ON is the NAMELIST information transfer initiated.

3.2.3 Procedure NRPNABL (NRPNABL). This CCL procedure, which is described more fully in ref. j, is the NORAPS counterpart of NGPOINT above. **Note:** as discussed in ref. a, the two procedures should probably be combined at some convenient time.

If/when this procedure is combined with procedure NGPOINT, sense switch two could be set ON by the interactive program to indicate that a NORAPS instead of a NOGAPS namelist is involved. This switch setting would then be used to initiate the appropriate file transfer since different procedures are used for that purpose - depending on which of the two larger-scale model systems is involved.

3.2.4 Program NGNPNT (NGNPNT). This interactive program is initiated by procedure NGPOINT above. By query/response it obtains the user's input with regard to which geographic points should be added to, deleted from or changed on the current DATAPOINT_DETAILS file. When the user has completed this dialogue, the programs writes out a new DATAPOINT_NAMELIST file (assuming there have been changes, or that it is an entirely new file) and sets the appropriate sense switch(es) to inform procedure NGPOINT what has happened.

This program CATALOGs the new/revised DATAPOINT_DETAILS file if one is created (the expected case). A table of the DATAPOINT_DETAILS and DATAPOINT_NAMELIST file data elements may be found in Appendix D.

Program NGNPNT satisfies the Get User Input requirement (paragraph 3.2.1.2 of ref. a) for NOGAPS.

Control of program execution is provided entirely by user responses.

The individual CSUs (modules) comprising the NGNPNT program are presented in Section 4.4.

3.2.5 Program NRPPNT (NRPPNT). This interactive program is the NORAPS counterpart of NGNPNT above.

This program differs from its NOGAPS counterpart above by writing an "intermediary" CCL procedure, PRO, which will pass on the identification of the particular NORAPS area/domain the user has specified. This is not necessary in the case of NOGAPS since there is only one (global) domain. A second difference is caused by the current (questionable) convention of not requiring users to input their assessment as to the "wet" or "dry" condition of each geographic point. A third difference is caused by the need to append at the beginning of each DATAPOINT_NAMELIST file a set of 20 integer, logical, or character variables which control global (programmatic) aspects of NORAPS model execution. These variables are unrelated to NABL and can vary from NORAPS domain to NORAPS domain.

The individual CSUs (modules) comprising the NRPPNT program are presented in Section 4.5.

Note: as discussed in ref. a, the two programs (NGNPNT and NRPPNT) should be combined at some convenient time. (See related **Note:** in paragraph 3.2.1.2 of ref. a and Appendix E hereto.)

3.2.6 Procedure NGNMOD (NGNMOD). This CCL procedure transfers a new or revised list of geographic points (DATAPOINT_NAMELIST) to SAM for use by NOGAPS. It ROUTEs a job (OKNMD) into SAM's input queue with the new data file attached to the job. That job will COPY and SWITCH the tag-a-long file to replace the older DATAPOINT_NAMELIST file.

Procedure NGNMOD satisfies the Transfer Points requirement (paragraph 3.2.1.3 of ref. a) for NOGAPS.

Control of procedure NGNMOD initiation is provided by procedure NGPOINT. Control during procedure execution is not required.

3.2.7 Procedure NRNMOD (NRNMOD). This CCL procedure is the NORAPS counterpart of NGNMOD above.

It differs from NGNMOD because it must be aware of which NORAPS area is to be changed and then pass this information on into SAM so that the correct area-corresponding namelist file can be updated. An integer area identifier is passed into this "permanent" procedure as a parameter when this procedure is called (BEGINed) from within the "intermediary" CCL procedure named PRO which was discussed above in paragraph 3.2.5.

Note: these two permanent transfer procedures (NGNMOD and NRNMOD) are substantially different. There would be no benefit if they were combined.

3.3 NABLS Interface descriptions. Figure 3-1 shows the external interfaces of the NABLS. Descriptions of each of those interfaces is provided in the following subparagraphs. (The information in this subsection draws heavily on material presented in subsection 3.1 of ref. a.)

3.3.1 USER_ONLINE_TERMINAL. This interface provides interactive input from the user who is prompted to provide details concerning the location and character of the data to be extracted from the (user-specified) larger-scale model.

The USER_ONLINE_TERMINAL interface will obtain the required information via a standard INPUT and OUTPUT connected "remote" terminal and the prompting program will write that information to a file in Fortran 200 NAMELIST format (as used on the Cyber 205, see ref. n) for later transfer and input to the appropriate larger-scale atmospheric prediction system.

This interface will use free-form query and response to obtain the USER_ONLINE_TERMINAL data elements presented in Appendix D.

3.3.2 NAMELIST_DATA. This interface consists of the NAMELIST input to the NABL Model. (NOTE: NAMELIST input/output is a CDC extension to ANSI-Standard FORTRAN and is approved for use at FNOC.)

The NAMELIST_DATA interface provides all of the unique details for a particular execution of the NABL model. The NAMELIST sequential data file is prepared off-line. It will be appended to the batch-mode job which causes execution of the NABL forecast model.

The data elements contained in the NAMELIST_DATA file are presented in Appendix D.

3.3.3 INITIAL_COND_DATA. This interface provides incoming initial conditions and hourly forecast tendency data from the larger-scale forecast model.

The INITIAL_COND_DATA interface provides access to NOGAPS or NORAPS initial condition data and forecast data for all of the points specified by the user through the USER_ONLINE_TERMINAL interface (paragraph 3.3.1) and passed to the larger-scale forecast system via the DATAPOINT_DETAILS interface (paragraph 3.3.4). The initial condition data resides on a formatted sequential file written by the larger-scale model during its forecast cycle.

The data elements contained in the INITIAL_COND_DATA file are presented in Appendix D.

3.3.4 DATAPOINT_DETAILS/DATAPOINT_NAMELIST. This closely related pair of interfaces provide and maintain the data point details (input by the NABLS user through the USER_ONLINE_TERMINAL interface (paragraph 3.3.1)) which must be read by the larger-scale forecast model (NOGAPS or NORAPS) in order to prepare the INITIAL_COND_DATA (paragraph 3.3.3) input to the NABL model.

The DATAPOINT_NAMELIST file is created on FED and "moved" to SAM. It should have a unique permanent file name which identifies it with a particular large-scale model domain. It is a formatted sequential file in FORTRAN 200 NAMELIST format (see ref. n).

The DATAPOINT_DETAILS file will reside on the front-end computer (FED) as a permanent sequential unformatted file. This file serves as the "master" which is read and updated (rewritten and CATALOGed) whenever a change to a DATAPOINT_NAMELIST file is required.

The master DATAPOINT_DETAILS files on the front-end and the DATAPOINT_NAMELIST input file to NOGAPS on SAM will contain only the data elements presented in Appendix D. For complicated reasons related to the developmental history of NORAPS and NABL, the DATAPOINT_NAMELIST files associated with NORAPS must have appended at their beginning a set of 20 integer, logical, or character variables which control global (programmatic) aspects of NORAPS model execution. These variables, which are unrelated to NABL and can vary from domain to domain, are not listed in the Appendix but are accounted for in program NRPPNT (paragraph 3.2.5) which now generates the NAMELIST file.

3.3.5 REF_EFFECTS_DATA. This interface provides a character (text) display of refractive effects for each point for each of the user-specified NABL forecast times (elements IPT and ITAU in the NAMELIST_DATA file).

The REF_EFFECTS_DATA interface provides sequential formatted file output suitable for direct input to any application program which may have use for such information (e.g. IREPS). Or, the file may be directly printed or used as message text.

This file is only generated when NAMELIST_DATA logical variable FNREFC is .TRUE.. It provides data at each point for each tau for all 40 of the NABL forecast levels.

On the following page is an example of an output file for one tau and point location.

```
NABL    12 HR FORECAST   VT   91082112Z AT LOCATION 32.4N 121.8W
INITIALIZED BY FNOC NOGAPS FIELDS            INCLUDES REFRACTION
40 LEVELS       SFC M-VAL=  362.9       TOPO=    32.6     WATER SFC
   PRES     TEMP  DEWPT        M        GRAD M        HT     REFRACTION
            (C)   DPRSN                 1000 FT      (FT)
  1016.8    17.1     3.1     344.2                   49.0    EVDT= 39FT
  1016.2    17.1     3.3     343.7     -28.905       65.4    TRAPPING
  1015.0    16.9     3.5     343.9       8.128       98.2    SUPER
  1012.6    16.7     3.5     345.6      24.924      163.9    NORMAL
  1009.7    16.4     3.5     348 3      32.607      245.9    NORMAL
  1005.5    16.1     3.3     352.4      35.844      360.7    NORMAL

            - etc. on up to levels 39 and 40 -

   607.4     2.9    26.4     859.3      42.395  14304.3      NORMAL
   596.2     1.7    25.4     880.4      42.806  14796.4      NORMAL
```

Note: "TOPO" in the third line is terrain height (ft). Also, "WATER SFC" would be "LAND SFC" if the point was considered "dry" (if its ground wetness value was less than 1.0).

3.3.6 **SKEWT_DATA.** This interface provides a character (text) display of the standard "vertical sounding" values required to prepare a "SKEWT" diagram based at each point for each of the user-specified NABL forecast times (elements IPT and ITAU in the NAMELIST_DATA file).

The SKEWT_DATA interface provides standard FNOC sequential formatted file output in "English" units suitable for direct input to any application program which may have use for such information (e.g. IREPS). Or, the file may be directly printed or used as message text.

This interface also provides a second slightly different "SKEWT" file in metric units which has been used DIAGNOSTICALLY by NOARL for input to a "SKEWT" plotting (graphics) program.

The first file is only generated when NAMELIST_DATA logical variable FNSKEW is .TRUE. and the second when NAMELIST variable NBSKEW is .TRUE. Both provide data at each point for each tau for all 40 of the NABL forecast levels.

On the following page are examples of the two types of output - the same one tau and point location for each type from the same forecast event - FNOC format first.

```
NABL    12 HR FORECAST   VT   91082112Z AT LOCATION 32.4N 121.8W
INITIALIZED BY FNOC NOGAPS FIELDS           INCLUDES WINDS
40 LEVELS                                            WATER SFC
    PRES      TEMP   DEWPT        DD        FF          HT
              (C)    DPRSN                             (FT)
  1016.8      17.1     3.1        325       16         49.0
  1016.2      17.1     3.3        325       16         65.4
  1015.0      16.9     3.5        325       17         98.2
  1012.6      16.7     3.5        325       17        163.9
  1009.7      16.4     3.5        325       18        245.9
  1005.5      16.1     3.3        326       18        360.7

          -  etc. on up to levels 39 and 40 -

   607.4       2.9    26.4        180        5      14304.3
   596.2       1.7    25.4        180        5      14796.4
```

Note: Column one (pressure) is in millibars, wind direction (DD) is in degrees true and wind speed (FF) is in knots. In the third line "WATER SFC" would read "LAND SFC" if the point was considered "dry" (if its ground wetness value was less than 1.0).
Also note: It is recommended that "TERRAIN HEIGHT xxx FT" be added to the third line as potentially useful information.

```
2
12789 91082100 12   32.4 -121.8 1 1 0 0 1 2 119.0  82.0 1.0000
                      * line two above cont'd *      9.4  .0  17.7
   1016.78   17.132   3.144    325.0    8.53      14
   1016.18   17.052   3.335    325.0    8.73      19
   1014.99   16.926   3.472    325.1    8.93      29
   1012.62   16.702   3.522    325.1    9.16      49
   1009.66   16.442   3.454    325.3    9.38      74
   1005.52   16.089   3.282    325.6    9.64     109

          -  etc. on up to levels 39 and 40 -

    607.61    2.867  26.449    179.7    2.72    4359
    596.23    1.676  25.448    179.7    2.72    4509
```

Note: Columns above are pressure (mb.), temperature (C), dewpoint depression (C), wind direction (deg. true), wind speed (mps), and NABL grid point height (m). Line one is the number of forecast times requested. Line two is block-station identifier, forecast base-DTG, forecast tau, latitude, longitude, five single-integer plot constants, model source indicator ("2" means NOGAPS), "I" coordinate of the data point, "J" coordinate, ground-wetness value (1.0000), terrain height (m), a land-or-sea flag (0 indicates a water point - see INITIAL_COND_DATA data element "PL_LRS" in Appendix D and read Appendix E for a complete description of this flag), and, last, surface temperature (C).

Also Note: A file nearly identical to the second (metric) file on the previous page is produced when NAMELIST_DATA variable NRSKEW is ".TRUE.". This third file contains SKEWT data based strictly on the INITIAL_COND_DATA from the large-scale model. It permits comparison with the NABL model output and is strictly DIAGNOSTIC.

3.3.7 COMPLETE_FORECAST_DATA. The COMPLETE_FORECAST_DATA interface will provide any (future) interfacing program a source file for any desired NABL forecast variable - assuming the resultant file is cataloged and made available for the appropriate applications.

The data elements to be contained in the COMPLETE_ FORECAST_ DATA file are presented in Appendix D.

Note: this interface is not provided for in the existing NABL code. It should consist of a compact sequential formatted file containing all of the forecast variables for each forecast point and tau. (There is provision within the existing code for extensive, but not compact, writes of many of the variables to standard OUTPUT. The existing OUTPUT is considered strictly DIAGNOSTIC.)

4. DETAILED DESIGN. This section describes the detailed design of each CSC. This is done by presenting and discussing each CSU (program module) associated with a particular CSC.

All procedure modules are written in CCL. A complete listing of each procedure is provided in Appendix B.

All program modules are written in ANSI-Standard FORTRAN using only FNOC-allowed non-ANSI features (the few exceptions to this rule are noted). A listing of the prologue for each program and subroutine is provided in Appendix C. Subroutine modules are presented alphabetically following their respective program modules - the same order as is used to present their prologues in Appendix C.

Hierarchy diagrams are provided for all programs and the associated calls are mentioned in the discussion. Calls to standard FNOC library subprograms, CDC system calls and uses of standard intrinsic functions are not documented. For such details the reader would be better served by the standard CDC Compiler Listings and Reference Maps.

4.1 Program NABL (NABL). Figure 4-1 shows the hierarchy of the twenty Program NABL modules.

4.1.1 Module NABL (Mod_NABL). This is the main driver (PROGRAM) "module" of the NABL "component" of the NABL "system".

Mod_NABL is larger and directly serves several more functions than would the normal "driver" in a more modularized program. In addition to initializing control variables and data arrays (based largely on PARAMETER and DATA statements within this main program and on information read from file NAMELIST_DATA), this module reads in all of the INITIAL_COND_DATA. The primary prediction and output cycles (taus within geographic points) are controlled from within this program based on the NAMELIST_DATA.

Program NABL calls 13 level-1 subroutines as shown in figure 4-1.

Note: It is suggested that the reading of INITIAL_COND_DATA (TAPE77) be placed into a new, separate subroutine. This could be done at the same time as provisions are made for the writing of a COMPLETE_FORECAST_DATA file as suggested in paragraph 3.3.7. Also at that time, the main point-by-point loop and the time-step-by-time-step computational loop within that first loop could both be moved out into a separate subroutine(s). These changes would add greatly to the modularity and therefore, presumably, to the maintainability of this complex code.

4.1.2 Subroutine BOUND (BOUND). This subroutine is called by program NABL. It computes the stability-dependent surface fluxes of heat, moisture and momentum; computes the lower-boundary turbulence variables; and calculates the evaporative duct height.

4.1.3 Subroutine CORCR (CORCR). This level-2 subroutine is called by subroutine CORRD. It calculates longwave upward and downward fluxes and heating rates and comparable values for shortwave (solar) radiation.

Subroutine CORCR calls subroutines CORWL and CORWS.

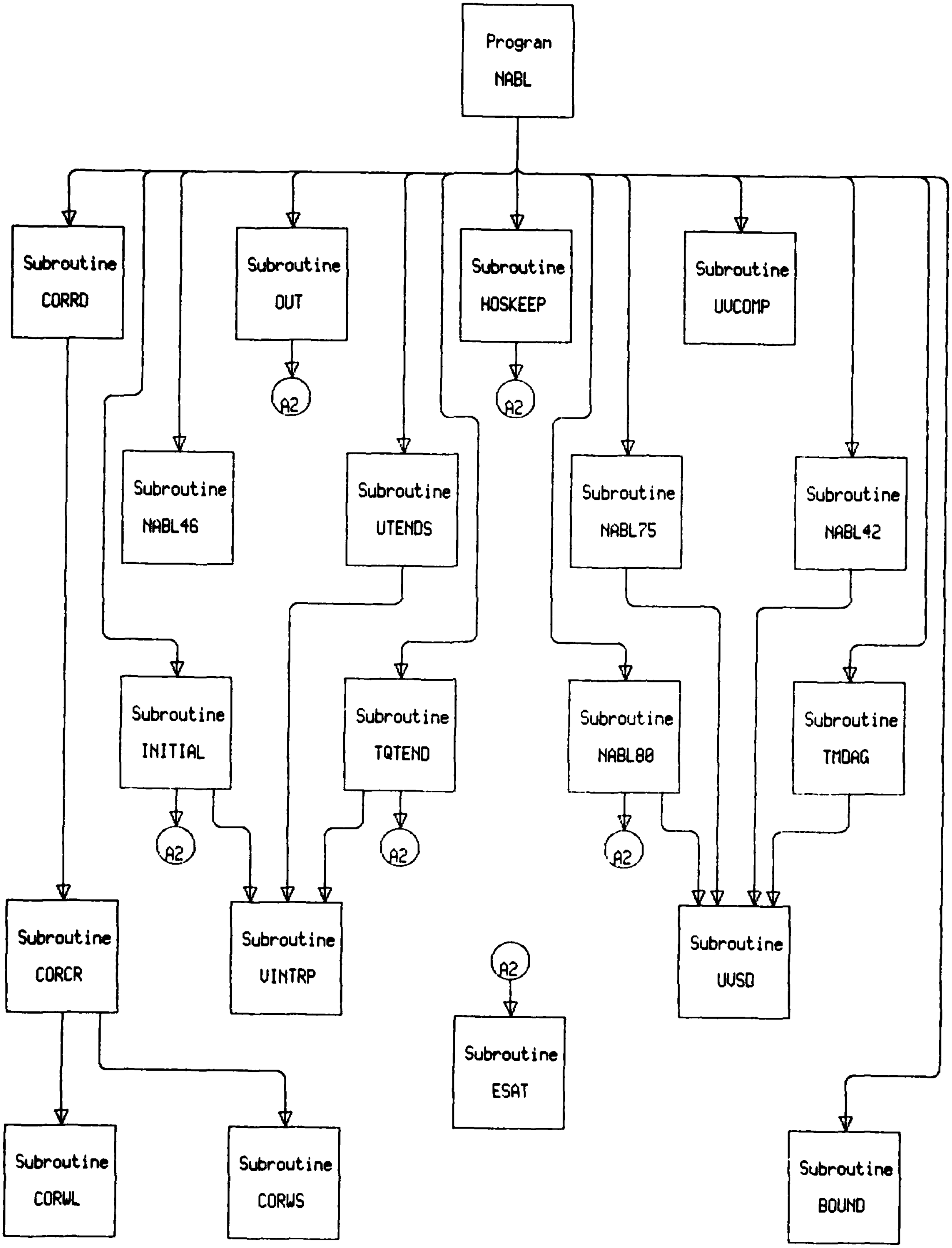

Figure 4-1. Program NABL hierarchy.

4.1.4 Subroutine CORRD (CORRD). This subroutine is called by program NABL. It is the main radiation subprogram. Integrated column abundances of water vapor and carbon dioxide are computed, the black body functions at each forecast level are calculated and solar flux at the top of the atmosphere is defined. Then fluxes are calculated by calling subroutine CORCR.

4.1.5 Subroutine CORWL (CORWL). This level-3 subroutine is called by subroutine CORCR. It computes the transmission functions which are needed to calculate <u>long-wave</u> radiative fluxes.

4.1.6 Subroutine CORWS (CORWS). This level-3 subroutine is called by subroutine CORCR. It computes the transmission functions which are needed to calculate <u>short-wave</u> radiative fluxes.

4.1.7 Subroutine ESAT (ESAT). This subroutine is called by five different level 1 subroutines (see figure 4-1). It computes the saturation vapor pressure (mb.) for a given temperature (K).

4.1.8 Subroutine HOSKEEP (HOSKEEP). This subroutine is called by program NABL. It calculates the grid volume liquid water content, the fractional cloudiness, and the eddy transfer coefficients.

Subroutine HOSKEEP calls subroutine ESAT.

4.1.9 Subroutine INITIAL (INITIAL). This subroutine is called by program NABL. It interpolates the initial condition data from the NOGAPS or NORAPS sigma-levels to the fixed-elevation points on the NABL grid. Also, several arrays used to store turbulence variables are initialized to zero and initial values for radiative flux transfers above the computational grid are calculated.

Subroutine INITIAL calls subroutines ESAT and VINTRP.

4.1.10 Subroutine NABL42 (NABL42). This subroutine is called by program NABL whenever NAMELIST_DATA logical variable FNSKEW is .TRUE.. It writes a SKEWT_DATA point-tau logical record to TAPE42 in FNOC standard (English units) format.

Subroutine NABL42 calls subroutine UVSD.

4.1.11 Subroutine NABL46 (NABL46). This subroutine is called by program NABL whenever NAMELIST_DATA logical variable FNREFC is .TRUE.. It writes a REF_EFFECTS_DATA point-tau logical record to TAPE46 in FNOC standard (English units) format.

4.1.12 Subroutine NABL75 (NABL75). This subroutine is called by program NABL whenever NAMELIST_DATA logical variable NBSKEW is .TRUE.. It writes a SKEWT_DATA point-tau logical record to TAPE75 in NOARL (metric units) format.

Subroutine NABL75 calls subroutine UVSD.

4.1.13 Subroutine NABL80 (NABL80). This subroutine is called by program NABL whenever NAMELIST_DATA logical variable NRSKEW is .TRUE.. It writes a SKEWT-type point-tau logical record to TAPE80 in NOARL (metric units) format. [Note: these records re-

flect the larger-scale model's output (as input to NABL) and not the results of the NABL model's forecast.]

Subroutine NABL80 calls subroutines ESAT and UVSD.

4.1.14 Subroutine OUT (OUT). This subroutine is called by program NABL. It writes the contents of nearly all forecast variables to standard OUTPUT (unless NAMELIST_DATA logical variable DSKIP is .TRUE., in which case this output is substantially abbreviated). **Note:** this subroutine could be replaced or largely rewritten to provide for the COMPLETE_FORECAST_DATA file output discussed in paragraph 3.3.7.

Subroutine OUT calls subroutine ESAT.

4.1.15 Subroutine TMDAG (TMDAG). This subroutine is called by program NABL only if logical NAMELIST_DATA variable TSER is .TRUE.. This routine loads selected NABL forecast variables into the large real array PDATA. Program NABL writes the PDATA array to TAPE70 for DIAGNOSTIC off-line "time series plotting" at the very end of program execution (but only if TSER is .TRUE.).

Subroutine TMDAG calls subroutine UVSD.

4.1.16 Subroutine TQTEND (TQTEND). This subroutine is called by program NABL. It computes the hourly average adiabatic tendencies for temperature and moisture based on the INITIAL_COND_DATA.

Subroutine TQTEND calls subroutines ESAT and VINTRP.

4.1.17 Subroutine UTENDS (UTENDS). This subroutine is called by program NABL. It computes the hourly average wind component tendencies based on the INITIAL_COND_DATA. The larger-scale model wind data are interpolated to the NABL grid points in this subprogram.

Subroutine UTENDS calls subroutine VINTRP.

4.1.18 Subroutine UVCOMP (UVCOMP). This subroutine is called by program NABL. It computes the momentum flux divergence for a given wind component.

4.1.19 Subroutine UVSD (UVSD). This subroutine is called by subroutines NABL42, NABL75, NABL80 and TMDAG. It converts U and V wind components to speed (m/s) and direction (deg.T).

4.1.20 Subroutine VINTRP (VINTRP). This level-2 subroutine is call by subroutines INITIAL, TQTEND and UTENDS. It interpolates values from the larger-scale NOGAPS or NORAPS sigma-levels to NABL vertical-grid coordinates.

4.2 Procedure NGPOINT (NGPOINT). This procedure executes interactive program NGNPNT and, if upon completion of that program sense-switch one is set ON (indicating that a new DATAPOINTS_NAMELIST file for NOGAPS use has been created), this procedure then executes (BEGINs) companion procedure NGNMOD.

This procedure is listed in Appendix B.

Note: It is suggested that this procedure be combined with procedure NRPNABL below.

4.3 Procedure NRPNABL (NRPNABL). This procedure executes interactive program NRPPNT and, if upon completion of that program sense-switch one is set ON (indicating that a new DATAPOINTS_NAMELIST file for NORAPS use has been created), this procedure then executes (BEGINs) intermediary procedure PRO.

This procedure and procedure PRO are listed in Appendix B.

Note: It is suggested that this procedure be combined with procedure NGPOINT above, using, for example, sense-switch two to indicate which follow-on transfer procedure (NGNMOD or PRO) should be executed.

4.4 Program NGNPNT (NGNPNT). Figure 4-2 shows the hierarchy of the nine Program NGNPNT modules.

Note: It is suggested that this program be combined with program NRPPNT below, this would eliminate subroutine redundancy in the FNOC source library, simplify program maintenance, and ensure a common user interface (which should include prompting for land-sea information for NORAPS). (See related note in paragraph 3.2.5 and Appendix E.)

4.4.1 Module NGNPNT (Mod_NGNPNT). This is the main driver (PROGRAM) "module" of the NGNPNT "component" of the NABL "system". In good modular fashion it does no more than connect INPUT and OUTPUT to the user's on-line terminal and then call level-1 subroutines to do the major work. Before exiting, this program sets sense switch one ON if a new DATAPOINT_NAMELIST file has been created.

Program NGNPNT calls five level-1 subroutines as shown in figure 4-2.

4.4.2 Subroutine NGEXTR (NGEXTR). This level-3 subroutine is called by subroutine NGLL. It extracts characters, integers or real numbers from character strings.

4.4.3 Subroutine NGFORM (NGFORM). This subroutine is called by program NGNPNT. It writes the user's point information in CDC FORTRAN 200 NAMELIST format to TAPE8 (DATAPOINT_NAMELIST file).

4.4.4 Subroutine NGLL (NGLL). This level-2 subroutine is called by subroutine NGRPT. It provides an interface to the on-line user for the purpose of obtaining valid point latitudes and longitudes and prompts for the user's land or ocean assessment of the point.

Subroutine NGLL calls subroutines NGEXTR, NGSTOP and NGYN.

4.4.5 Subroutine NGOLD (NGOLD). This subroutine is called by program NGNPNT. At the top this subroutine reads existing (old) point information from TAPE6 (file NGNREC) into internal arrays. At alternate entry point NGWRT it writes the, presumably new or revised, point information back out to a new TAPE6 (NGNREC file).

Subroutine NGOLD calls subroutine NGSTOP.

Note: Entry point NGWRT is depicted on figure 4-2 as a separate subroutine. It is recommended that it actually be made so.

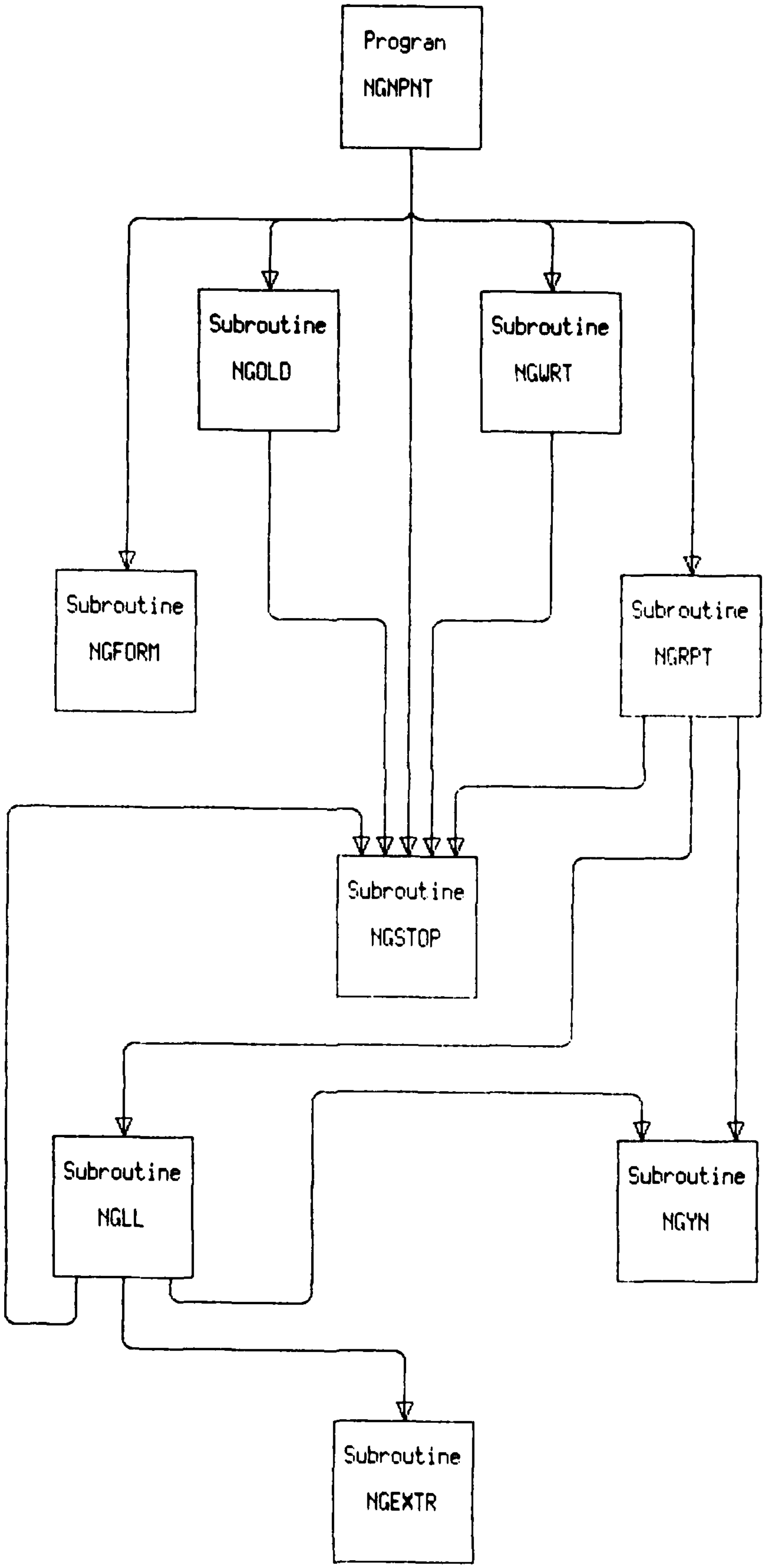

Figure 4-2. Program NGNPNT hierarchy.

4.4.6 Subroutine NGRPT (NGRPT). This subroutine is called by program NGNPNT. It conducts the query-response dialog with the user - prompting that person to enter or reenter point information details.

Subroutine NGRPT calls subroutines NGLL, NGSTOP and NGYN.

4.4.7 Subroutine NGSTOP (NGSTOP). This level-1,2,3 subroutine provides a common program exit for both normal and abnormal terminations.

4.4.8 Subroutine NGWRT (NGWRT). This is currently an entry point in subroutine NGOLD. See paragraph 4.4.5 above.

This entry calls subroutine NGSTOP.

4.4.9 Subroutine NGYN (NGYN). This level-2,3 subroutine prompts the user for a yes or no response.

Note: This subroutine uses "alternate returns" for yes or no responses. This technique is not permitted by FNOC programming standards (ref. k). It is recommended a logical variable return or something similar be employed when/if program NGNPNT is revised. Such a change would require minor changes to this subroutine and the two subroutines which call it (NGLL and NGRPT).

4.5 Program NRPPNT (NRPPNT). Figure 4-3 shows the hierarchy of the ten Program NRPPNT modules.

Note: It is suggested that this program be combined with program NGNPNT above, this would eliminate subroutine redundancy in the FNOC source library, simplify program maintenance, and ensure a common user interface (which should include prompting for land-sea information for NORAPS). (See related note in paragraph 3.2.5 and Appendix E.)

4.5.1 Module NRPPNT (Mod_NRPPNT). This is the main driver (PROGRAM) "module" of the NRPPNT "component" of the NABL "system". In good modular fashion it does no more than connect INPUT and OUTPUT to the user's on-line terminal and then call level-1 subroutines to do the major work. Before exiting this program sets sense switch one ON if a new DATAPOINT_NAMELIST file has been created.

Program NGNPNT calls six level-1 subroutines as shown in figure 4-3.

4.5.2 Subroutine NREXTR (NREXTR). This level-3 subroutine is called by subroutine NRLL. It extracts characters, integers or real numbers from character strings.

4.5.3 Subroutine NRFORM (NRFORM). This subroutine is called by program NRPPNT. It writes the user's point information in CDC FORTRAN 200 NAMELIST format to TAPE8 (the DATAPOINT_NAMELIST file).

As previously discussed, this subroutine appends at the beginning of the NAMELIST a set of 20 variables which are unrelated to NABL. These variables are required to control NORAPS model execution, and may vary from one forecast area (NORAPS domain) to another. The values for these variables (stored in

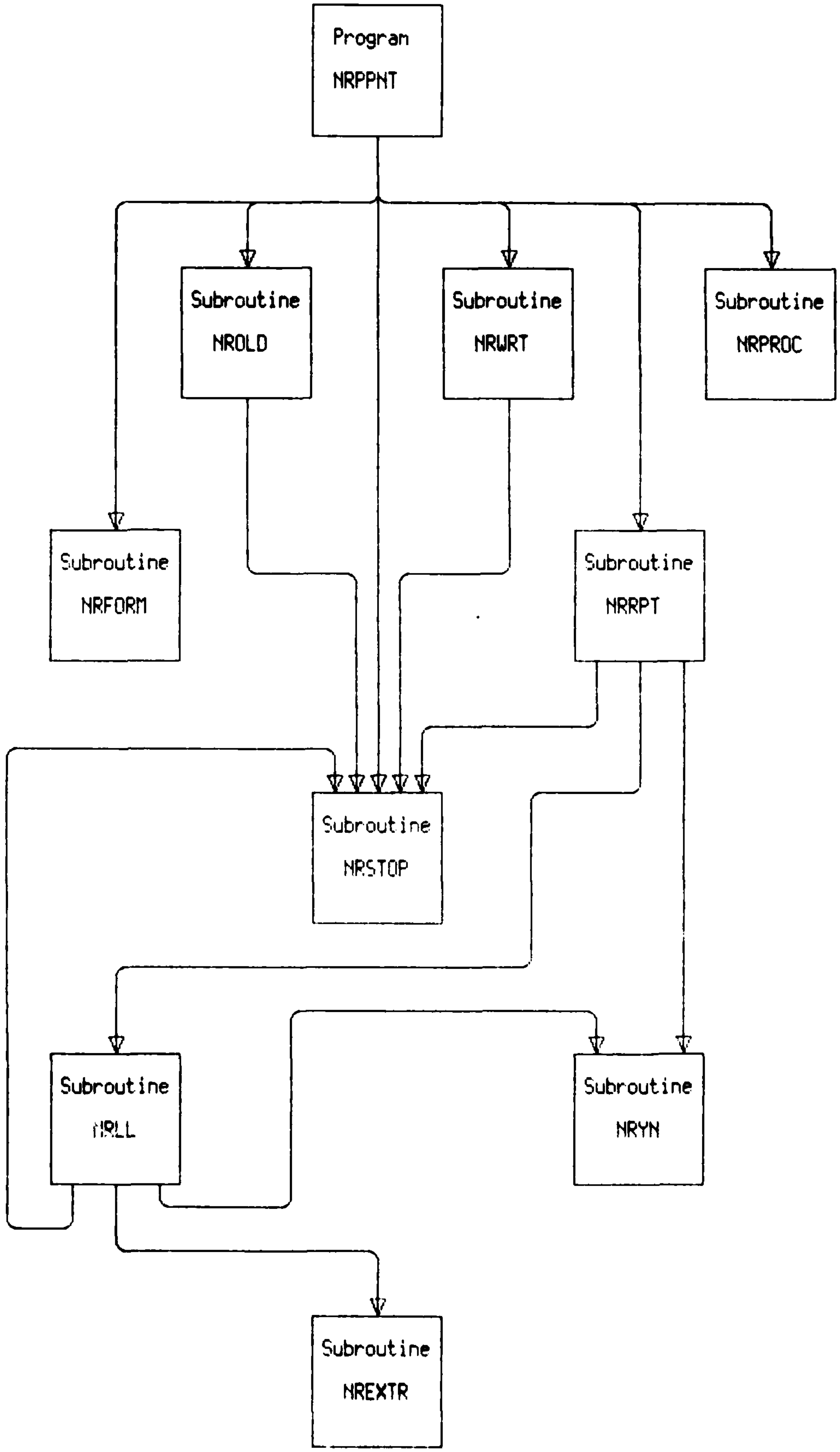

Figure 4-3. Program NRPPNT hierarchy.

local arrays initialized by DATA statements) must be obtained from the person(s) responsible for NORAPS system maintenance.

4.5.4 Subroutine NRLL (NRLL). This level-2 subroutine is called by subroutine NRRPT. It provides an interface to the on-line user for the purpose of obtaining valid point latitudes and longitudes.

Subroutine NRLL calls subroutines NREXTR, NRSTOP and NRYN.

4.5.5 Subroutine NROLD (NROLD). This subroutine is called by program NRPPNT. At the top this subroutine reads existing (old) point information from TAPE6 (file NRNREC) into internal arrays. At alternate entry point NRWRT it writes the, presumably new or revised, point information back out to a new TAPE6 (NRNREC file).

Subroutine NROLD calls subroutine NRSTOP.

Note: Entry point NRWRT is depicted on figure 4-3 as a separate subroutine. It is recommended it actually be made so.

4.5.6 Subroutine NRPROC (NRPROC). This subroutine is called by program NRPPNT. It writes out the short intermediary procedure PROC which is used to pass the NORAPS area identifier on into procedure NRNMOD. Procedure PRO is shown in Appendix B.

Note: NRPROC is the only subroutine in program NRPPNT which does not have a nearly exact counterpart in program NGNPNT.

4.5.7 Subroutine NRRPT (NRRPT). This subroutine is called by program NRPPNT. It conducts the query-response dialog with the user - prompting that person to enter or reenter point information details.

Subroutine NRRPT calls subroutines NRLL, NRSTOP and NRYN.

4.5.8 Subroutine NRSTOP (NRSTOP). This level-1,2,3 subroutine provides a common program exit for both normal and abnormal terminations.

4.5.9 Subroutine NRWRT (NRWRT). This is currently an entry point in subroutine NROLD. See paragraph 4.5.5 above.

This entry calls subroutine NRSTOP.

4.5.10 Subroutine NRYN (NRYN). This level-2,3 subroutine prompts the user for a yes or no response.

Note: Like its counterpart, NGYN, this subroutine uses "alternate returns" for yes or no responses. This technique is not permitted by FNOC programming standards (ref. k). It is recommended a logical variable return or something similar be employed when/if program NRPPNT is revised and/or combined with program NGNPNT. Such a change would require minor changes to this subroutine and the two subroutines which call it (NRLL and NRRPT).

4.6 Procedure NGNMOD (NGNMOD). This procedure transfers a list of points (DATAPOINT_NAMELIST) to SAM and replaces file DGFILE on SAM with the new file. The job routed to SAM's input queue by this procedure is OKNMD. That job carries with it into SAM a copy of the DATAPOINT_NAMELIST file. The SAM job runs with

operational priority and requires less than 1 second CPU time.

This procedure is listed in Appendix B.

4.7 Procedure NRNMOD (NRNMOD). This procedure is begun by executing intermediary procedure PRO. This procedure transfers a list of points (DATAPOINT_NAMELIST) to SAM and replaces file NRNFILn (where integer n identifies the NORAPS area) with the new file. The job routed to SAM's input queue by this procedure is OKNRD. That job carries with it into SAM a copy of the DATAPOINT_NAMELIST file. The SAM job runs with operational priority and requires less than 1 second CPU time.

This procedure is listed in Appendix B.

5. NABLS DATA. This section is expected to describe the global data elements within the NABLS. All such elements are contained in the DATAPOINT_DETAILS file and/or the NAMELIST_DATA file. Both of these files are described in Appendix D.

6. NABLS Data Files. This section is expected to describe each of the data files of the NABLS which are shared by more than one component or module of the system. Those files would be the namelist-format NAMELIST_DATA file, the unformatted sequential DATAPOINT_DETAILS file and the formatted INITIAL_COND_FILE. All three of these files are described in Appendix D. (Note: these files are really shared through the medium of COMMONs into which their contents are read. See the NABL prologue in Appendix C.)

7. REQUIREMENTS TRACEABILITY. The following table relates paragraphs in the NABLS requirements document (ref. a) to the corresponding/supporting paragraphs in this document.

Requirement (reference a. paragraph)	Design Specification (paragraph herein)
3.1 NABLS external interface requirements	
3.1.1 USER_ONLINE_TERMINAL	3.3.1
3.1.2 NAMELIST_DATA	3.3.2
3.1.3 INITIAL_COND_DATA	3.3.3
3.1.4 DATAPOINT_DETAILS/DATAPOINT_NAMELIST	3.3.4
3.1.5 REF_EFFECTS_DATA	3.3.5
3.1.6 SKEWT_DATA	3.3.6
3.1.7 COMPLETE_FORECAST_DATA	3.3.7
3.2 NABLS capability requirements	
3.2.1 Interface Larger-scale Models	
3.2.1.1 Interface Control	4.2 & 4.3
3.2.1.2 Get User Input	4.4 & 4.5
3.2.1.3 Transfer Points	4.6 & 4.7
3.2.1.4 Transfer Model Output	(external to NABLS)
3.2.2 Process Input Data	4.1
3.2.3 Compute Forecast Variables	4.1
3.2.4 Prepare Output	4.1
3.3 NABLS internal interfaces	
3.3.1 Initiate User Input	4.2 & 4.3
3.3.2 Convey Datapoint Details	4.4 & 4.5
3.3.3 Initiate Datapoint Transfer	4.2 & 4.3
3.3.4 Convey Output Requirements	4.1.1
3.3.5 Convey Larger-Scale Data	4.1.1
3.3.6 Convey Forecast Data	4.1.1 & 4.1.10-15
3.3.7 Control Model Execution	4.1.1
3.4 NABLS data element requirements (internal).	
3.4.1 SENSE_SWITCH	4.4.1 & 4.5.1
3.4.2 NORAPS_PROC	4.5.6 & 4.7
3.4.3 DATAPOINT_NAMELIST	4.4.3 & 4.5.3
3.4.4 OUTPUT_CONTROL_FLAGS	4.1.1
3.4.5 LARGER_SCALE_DATA	4.1.1
3.4.6 FORECAST_VARIABLES	4.1.1 & 4.1.10-15

8. Notes.

8.1 Acronyms, abbreviations and identifiers.

This appendix defines acronyms and abbreviations used in the document text. It also provides a long-title and paragraph reference for each project unique identifier (except for subroutine modules).

Short Title	Long Title and/or Explanation
CCL	(CDC) Cyber (job) Control Language
CDC	Control Data Corporation
COMPLETE_FORECAST_DATA	Complete Forecast Data External Interface/File (para. 3.1.7)
CSC	Computer Software Component (subset of a CSCI)
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit (subset of a CSC)
DATAPOINT_DETAILS	Datapoint Details External Interface/File and Internal Data Set (para. 3.3.4)
DATAPOINT_NAMELIST	Datapoint Namelist Internal Interface/File (para. 3.3.4)
DFD	Data Flow Diagram
FED	The designated "Front-End" computer (usually a CDC Cyber 860)
FNOC	Fleet Numerical Oceanography Center, Monterey, CA
IDD	Interface Design Document
INITIAL_COND_DATA	Initial Conditions Data External Interface/File (para. 3.3.3)
IREPS	Integrated Refractive Effects Prediction System
IRS	Interface Requirements Specification
NABL	Naval Atmospheric Boundary Layer (model or program)
NABLS	NABL Model System
NAMELIST_DATA	Namelist Data External Interface/File (para. 3.1.2)
NGNMOD	NOGAPS Transfer Points Procedure (para. 3.2.6 and 4.6)

Short Title	Long Title and/or Explanation
NGNPNT	NOGAPS Get User Input Program (para. 3.2.4 and 4.4)
NGPOINT	NOGAPS Interface Control Procedure (para. 3.2.2 and 4.2)
NOARL	Naval Oceanographic and Atmospheric Research Laboratory
NOGAPS	Navy Operational Global Atmospheric Prediction Model
NORAPS	Navy Operational Regional Atmospheric Prediction Model
NOS/BE	(CDC's) Network Operating System/Batch Environment
NRNMOD	NORAPS Transfer Points Procedure (para. 3.2.7 and 4.7)
NRPNABL	NOGAPS Interface Control Procedure (para. 3.2.3 and 4.3)
NRPPNT	NORAPS Get User Input Program (para. 3.2.5 and 4.5)
REF_EFFECTS_DATA	Refractive Effects Data External Interface/file (para. 3.3.5)
SAM	The CDC Cyber 205 "super-computer" System
SDD	Software Design Document
SKEWT_DATA	SKEWT Data External Interface/File (para. 3.3.6)
SRS	Software Requirements Specification
USER_ONLINE_ TERMINAL	User Online Terminal External Interface (para. 3.3.1)
VSOS	(CDC's) Virtual Storage Operating System

Appendixes.

Appendix A. Data flow diagram (DFD) symbology

A data flow diagram (DFD) is a graphical technique used to depict information flow and the transforms that are applied as data move from input to output. It is also known as a bubble chart. The following illustrates the basic form of a DFD as utilized in this document.

External Entity	A source of system inputs, or sink of system outputs.
Process	Performs some transformation of its input data to yield its output data.
Data Flow ——————>	Shows the data items moving into or out of processes, data stores, sources or sinks; the arrowhead indicates the direction of data transfer.
Control Flow - - - - - - - - >	Shows the exercise of control of data flow from one exercised by one external entity or process upon another; the arrowhead indicates direction of control transfer.
Data Store	A repository of data, in a logical sense; it can be implemented as a disk file, a memory buffer, or a database, etc.

Appendix B. Procedure File Listings

Listing of Procedure NGPOINT.

```
NGPOINT.
LIBRARY(*FNWCLIB)
APLIB(,*KEEP)
NDF,OFF.
KEEP(FNWCLIB)
ETL,100.
MOUNT,SN=SHARDSK.
APLIB(MT1731,*NGNPNT)
NDF,ON.
OFFSW(1)
NGNPNT.
RETURN,TAPE7,NGNPNT.
COMMENT. TEST FOR AND RUN PRO HERE
IFX(SSW,1,NOSAM)
NDF,OFF.
REWIND,TAPE8.
COPYSBF,TAPE8,OUTPUT.
REWIND,TAPE8.
NDF,ON.
BEGIN,NGNMOD,OPSFIL.
ENDIF(NOSAM)
DISCONT,INPUT.
DISCONT,OUTPUT.
NDF,OFF.
APLIB(,*KEEP)
KEEP(FNWCLIB)
DSMOUNT,SN=SHARDSK.
NDF,ON.
REVERT.
EXIT(P)
DISCONT(INPUT)
DISCONT(OUTPUT)
APLIB(,*KEEP)
KEEP(FNWCLIB)
DSMOUNT,SN=SHARDSK.
NDF,ON.
REVERT.
EXIT.
REVERT(ABORT)
EXIT.
EXIT.
C   INTERFACE PROCEDURE TO SET NEW NABL LOCATIONS ON SAM FILE
C    DGFILE FOR NOGAPS.
C    LEWIT  CSC  12/90
```

Listing of Procedure NRPNABL.

```
NRPNABL.
LIBRARY(*FNWCLIB)
APLIB(,*KEEP)
NDF,OFF.
KEEP(FNWCLIB)
ETL,100.
MOUNT,SN=SHARDSK.
APLIB(MT1731,*NRPPNT)
NDF,ON.
OFFSW(1)
NRPPNT.
COMMENT. TEST FOR AND RUN PRO HERE
IFX(SSW,1,NOSAM)
NDF,OFF.
REWIND,TAPE8.
COPYSBF,TAPE8,OUTPUT.
REWIND,TAPE8.
NDF,ON.
REWIND,PRO.
BEGIN,PRO,PRO.
ENDIF(NOSAM)
DISCONT,INPUT.
DISCONT,OUTPUT.
NDF,OFF.
APLIB(,*KEEP)
KEEP(FNWCLIB)
DSMOUNT,SN=SHARDSK.
NDF,ON.
REVERT.
EXIT(P)
DISCONT(INPUT)
DISCONT(OUTPUT)
APLIB(,*KEEP)
KEEP(FNWCLIB)
DSMOUNT,SN=SHARDSK.
NDF,ON.
REVERT.
EXIT.
REVERT(ABORT)
EXIT.
EXIT.
C   INTERFACE PROCEDURE TO SET NEW NABL LOCATIONS ON SAM FILE
C    NRNFIL7 FOR NOR..S.
C    LEWIT  CSC  12/8.
```

Listing of Procedure NGNMOD.

```
NGNMOD.
COPYBR,SAMJOB,SAMIN,1.
COPYBF,TAPE8,SAMIN.
RETURN,SAMJOB.
REWIND,SAMIN.
ROUTE,SAMIN,DC=IN.
REVERT.
EXIT.
REVERT,ABORT.
EXIT.
EXIT.
 PROCEDURE TO MAKE NEW VERSION OF NABL INPUT FILE DGFILE ON
   SAM FOR NOGAPS.  INITIATED BY PROCEDURE NGPOINT.
CHANGE RECORD
<<CHANGE NOTICE>> NGNMOD*01 ( 20 MARCH 1991) -- LEWIT
  ROUTPUT OUTPUT TO BIN 46
 CREATED BY H. LEWIT CSC 12/90
/DATA SAMJOB
OKNMD,STSAM.46 LEWIT
USER(U=xxxxxx,AC=xxxxxxx)
RESOURCE(TL=100,PRIO=12,WS=1000,LP=5,JCAT=OPS)
ROUTPUT(DC=PR,TID=C,BIN=46,S=0,CO=$DGFILE CHG$)
COPY,INPUT,DGHOLD.
COPY,DGHOLD,OUTPUT.
PURGE,DGFILE.
RETURN,DGFILE.
SWITCH,DGHOLD,DGFILE.
DEFINE,DGFILE.
PERMIT,DGFILE,USER=*,ACCESS=R.
COPY,DGFILE,OUTPUT.
EXIT.
```

Listing of Procedure PRO.

```
PRO.
BEGIN,NRNMOD,OPSFIL,n.
REVERT.
```

Note: the integer "n" in line two above identifies a particular NORAPS area.

Listing of Procedure NRNMOD.

```
NRNMOD,NAREA.
COPYBR,SAMJOB,SAMIN,1.
COPYBF,TAPE8,SAMIN.
RETURN,SAMJOB.
REWIND,SAMIN.
ROUTE,SAMIN,DC=IN.
REVERT.
EXIT.
REVERT,ABORT.
EXIT.
EXIT.
 PROCEDURE TO MAKE NEW VERSION OF NABL INPUT FILE
NRNFIL_NAREA ON
   SAM FOR NORAPS.  INITIATED BY PROCEDURE NRPNABL.
CHANGE RECORD
 CREATED BY H. LEWIT CSC 12/90
/DATA SAMJOB
OKNRD,STSAM.46 LEWIT
USER(U=xxxxxx,AC=xxxxxxx)
RESOURCE(TL=100,PRIO=12,WS=1000,LP=5,JCAT=OPS)
COPY,INPUT,NRNFIL.
PURGE,NRNFIL_NAREA.
SWITCH,NRNFIL,NRNFIL_NAREA.
DEFINE,NRNFIL_NAREA.
PERMIT,NRNFIL_NAREA,USER=*,ACCESS=R.
COPY,NRNFIL_NAREA,OUTPUT.
EXIT.
```

Appendix C. Module Proloques

```
      PROGRAM NABL
C
C...................START PROLOGUE..................................
C
C PROGRAM NAME:   NABL
C
C DESCRIPTION:
C
C     THIS PROGRAM INITIALIZES A ONE-DIMENSIONAL BOUNDARY LAYER
C     MODEL WITH PROFILES OF WIND, TEMPERATURE, AND MOISTURE
C     FROM NOGAPS OR NORAPS AT PRESELECTED LOCATIONS.  IT FORECASTS
C     THE EVOLUTION OF THESE PROFILES DUE TO PARAMETERIZED PHYSICAL
C     PROCESSES, SURFACE FORCING, AND SYNOPTIC-SCALE ADIABATIC,
C     ADVECTIVE TENDENCIES.  HIGH-RESOLUTION FORECASTS OF BOUNDARY
C     LAYER STRUCTURE OUT TO 24- 36 H ARE PRODUCED.
C
C ORIGINAL PROGRAMMER, DATE:    STEPHEN BURK (NOARL) 1987
C
C CURRENT PROGRAMMER:           ROLF LANGLAND (NOARL)
C                               HOWARD LEWIT (CSC)
C                               LIANA ZAMBRESKY (FNOC)
C
C COMPUTER/OPERATING SYSTEM:    CYBER 205 VSOS
C
C CLASSIFICATION:               UNCLASSIFIED
C
C INPUT FILES:
C
C      NAMELIST = INPUT FROM POOL FILE CONTAINING DIRECTIVES FOR
C                 LOCATIONS AND LENGTH OF NABL FORECASTS (TAPE8)
C
C      NLTAUF    - FINAL TAU OF NABL FORECAST
C      NTAUF     - # OF OUTPUT TAU(S) FOR NABL
C      ITAU      - TIME (IN HOURS) FOR NABL FORECAST OUTPUT
C      NFCST     - NUMBER OF LOCATIONS FOR NABL FORECASTS
C      IPT,JPT   - PAIRS OF I,J INPUT GRID POINTS
C      ALAT,ALON - LATITUDE/LONGITUDE(0-360 W) INPUT
C                  USE 1/10 DEG ACCURACY, AS IN NORAPS INPUT
C      ITYPE     - "IJ" (IPT,JPT) OR "LL" (ALAT,ALON) INPUT TYPE
C      NBLKST    - 5-DIGIT BLOCKSTATION IDENTIFIER (OPTIONAL)
C      NBSKEW    - TRUE IF NABL SKEW-T PLOTS DESIRED
C      NRSKEW    - TRUE FOR NORAPS SKEW-T PLOTS (AT 12-HR INTERVALS)
C      FNSKEW    - TRUE FOR FNOC FORMAT SKEWTS
C      FNREFC    - TRUE FOR FNOC REFRACTIVITY FILE
C      NRTAU     - TAU AT WHICH NORAPS SKEW-T WANTED
C      NSKEW     - # OF NORAPS SKEW-T TIMES WANTED
C      DSKIP     - TRUE IF LIMITED # OF FIELDS SENT TO OUTPUT
C      TSER      - TIME SERIES DIAGNOSTIC OPTION
C
C      TAPE77 = INPUT FILE CONTAINING INITIAL VALUES AND ADIABATIC
C               TEMPERATURE AND MOISTURE TENDENCIES FROM NORAPS.
C
C OUTPUT FILES:
C
```

```
C     TAPE42 = FORECAST DATA FOR SKEWT IN FNOC FORMAT
C     TAPE46 = FORECAST REFRACTIVITY VALUES
C     TAPE75 = NABL FORECAST SKEWT FILE
C     TAPE80 = NORAPS FORECAST SKEWT FILE
C     TAPE70 = PDATA - TIME SERIES DIAGNOSTICS
C
C..................MAINTENANCE SECTION.............................
C
C BRIEF DESCRIPTION OF PROGRAM MODULES:
C
C     INITIAL - SET UP INITIAL NABL PROFILE FROM NORAPS/NOGAPS INPUT
C     BOUND   - SURFACE FLUXES, TKE EQUATION, EVAP DUCT CALCULATION
C     HOSKEEP - EDDY TRANSFER COEFFICIENTS, SFC ENERGY BUDGET
C             - FRACTIONAL CLOUDINESS
C     UVCOMP  - MOMENTUM FLUX DIVERGENCE, NUDGING OF WIND COMPONENTS
C     CORRD   - MAIN RADIATION SUBROUTINE
C     CORCR   - RADIATIVE HEATING RATES
C     CORWL   - LONGWAVE TRANSMISSION FUNCTIONS
C     CORWS   - SHORTWAVE TRANSMISSION FUNCTIONS
C     BLKDATA - INITIALIZE CONSTANTS AND VARIABLES
C     VINTRP  - VERTICAL INTERPOLATIONS
C     ESAT    - SATURATION VAPOR PRESSURE
C     OUT     - PRINT NABL FORECAST OUTPUT
C     UTENDS  - COMPUTE WIND TENDENCIES FROM NORAPS/NOGAPS INPUT
C     TQTEND  - COMPUTE TEMPERATURE & MOISTURE TENDENCES
C     UVSD    - CONVERT U & V WIND.COMPONENTS TO SPEED/DIRECTION
C     NABL75  - WRITE NABL FORECAST TO FILE FOR SKEW-T PLOT
C     NABL80  - WRITE NORAPS FORECAST TO FILE FOR SKEW-T PLOT
C               (NOT AVAILABLE IN OPERATIONAL RUN)
C     NABL42  - WRITE NABL FORECAST TO FILE IN FNOC FORMAT
C     NABL46  - WRITE NABL REFRACTIVITY FORECAST TO OUTPUT FILE
C     TMDAG   - WRITE TO OUTPUT FILE FOR TIME SERIES PLOT
C               (FOR DIAGNOSTIC PURPOSES ONLY)
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C  ABT     = ABSOLUTE TEMP.(KELVIN)
C  ABTH    = POTENTIAL TEMP.
C  DT      = TIME STEP(SECONDS)
C  NN      = TOTAL NO. OF GRID PTS.
C  IT      = ITERATION NO.(COUNTER)
C  ITOT    = LAST ITERATION OF NABL FORECAST
C  P(I)    = PRESSURE(MB)
C  QL      = LIQUID WATER CONTENT(NONDIMENSIONAL,N.D.)
C  QW      = TOTAL WATER CONTENT(N.D.)
C  QWZ     = SURFACE(SFC) QW VALUE
C  SPECHUM = SPECIFIC HUMIDITY
C  TH      = VIRTUAL POTENTIAL TEMP.
C  THL     = LIQUID WATER POTENTIAL TEMP.(KELVIN)
C  THLZ    = SFC VALUE OF THL
C  U       = U-COMPONENT OF WIND(X-DIRECTION)
C  UG      = X-COMPONENT OF GEOSTROPHIC WIND
C  V       = Y-COMPONENT OF WIND
C  VG      = Y-COMP. OF GEOSTROPHIC WIND
C  XMZ     = SFC XM VALUE
```

```
C  Z(I)     = GRID PT. HEIGHTS
C  Z1       = SFC ROUGHNESS HT.
C
C COMMON BLOCKS:
C
C      BLANK COMMON - SEE BELOW.
C      ECSBLK - EXTENDED CORE MEMORY FOR NORAPS/NOGAPS TENDENCIES
C      BLK2   - ARRAYS CONTAINING GRID POINT INITIAL CONDITIONS
C               AND SPECIFICATIONS
C      BLK3   - ARRAYS CONTAINING HOURLY TENDENCY INFORMATION
C
C LANGUAGE:        FTN5
C
C RECORD OF CHANGES:
C
C...................END PROLOGUE.....................................
C
C
C   BLANK COMMON DESCRIPTION:
C
C     THE BLANK COMMON BLOCK CONTAINS MANY OF THE MAJOR MODEL
C     VARIABLES.  THE ARRAYS ARE DIMENSIONED ACCORDING TO THE
C     NUMBER OF MODEL GRID POINTS IN THE VERTICAL.  VARIABLE
C     NAMES ARE USUALLY CHOSEN TO BE SUGGESTIVE OF THE
C     MATHEMATICAL FORM THE EXPRESSION TAKES IN AN EQUATION.
C     FOR EXAMPLE, THE SCALING PARAMETERS OF MOMENTUM, POTENTIAL
C     TEMPERATURE, AND TOTAL MOISTURE ARE WRITTEN AS USTAR,
C     THLSTAR, AND QWSTAR, RESPECTIVELY --- SINCE THESE QUANTITIES
C     ARE WRITTEN MATHEMATICALLY WITH THE SUBSCRIPT "STAR".  AS
C     ANOTHER EXAMPLE, THE VARIABLE NAME FOR VIRTUAL HEAT FLUX
C     IS THVWB BECAUSE THIS QUANTITY GENERALLY IS WRITTEN
C     MATHEMATICALLY AS THE PRODUCT OF FLUCTUATIONS IN VIRTUAL
C     TEMPERATURE (THV) AND VERTICAL VELOCITY (W), WITH AN
C     OVERBAR (B) --- HENCE, THE EXPRESSION THVWB.

      PARAMETER (NN=40,NR=21,NL=11,NT=37,NP=20)

C        NN = NUMBER OF NABL VERTICAL LEVELS
C        NR = NUMBER OF NORAPS LEVELS
C        NL = NUMBER OF TENDENCY LEVELS
C        NT = LENGTH OF FORECAST IN HOURS + 1 (TAU 0)
C        NP = NUMBER OF OUTPUT POINTS
```

```
      SUBROUTINE BOUND

C........................START PROLOGUE...............................
C
C SUBPROGRAM NAME:   BOUND
C
C DESCRIPTION:
C     THIS SUBPROGRAM COMPUTES THE STABILITY-DEPENDENT SURFACE
C     FLUXES OF HEAT, MOISTURE, AND MOMENTUM.  THE ALGORITHMS
C     DESCRIBED BY LOUIS, BOUND. LAYER METEOR., 17, PP.187-202,
C     1979 ARE USED.  ALSO, LOWER BOUNDARY CONDITIONS ON TURBULENCE
C     VARIABLES ARE COMPUTED BY ASSUMPTION OF A PRODUCTION-
C     DISSIPATION EQUILIBRIUM IN THE TKE EQUATION.  EVAPORATIVE
C     DUCT HEIGHT CALCULATIONS ARE ALSO PERFORMED HERE.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:   ROLF LANGLAND (NOARL)
C                       HOWARD LEWIT (CSC)
C                       LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):  CALL BOUND
C
C........................MAINTENANCE SECTION..........................
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C........................END PROLOGUE.................................
```

```
      SUBROUTINE CORCR(QR,FPO,FMH,TR,X1U,X2U,GAM2,EO,EH,
     1                 A,PRATIO,QL,IM1,Z,C1,C2,C3,WK,L2,ENU,LL,
     2                 FM,FP,FN)
C
C.......................START PROLOGUE.............................
C
C SUBPROGRAM NAME:   CORCR
C
C DESCRIPTION:
C
C     THIS SUBPROGRAM IS USED IN SEPARATE CALLS TO COMPUTE,
C     (1) LONGWAVE UPWARD AND DOWNWARD FLUXES AND HEATING
C     RATES, AND THEN (2) THE COMPARABLE VALUES FOR SHORTWAVE
C     (SOLAR) RADIATION.  THE TRANSMISSION FUNCTION, WHICH
C     IS A FUNCTION OF THE INTEGRATED VAPOR, ETC., IN THE
C     COLUMN, IS COMPUTED BY CALLS TO THE SUBROUTINES
C     CORWL AND CORWS.  NOTE THAT CORWL AND CORWS ARE
C     EQUIVALENT WITH "TR" THROUGH THE ARGUMENT LIST.  AT CLOUD
C     TOP, A SPECIAL ONE-SIDED DIFFERENCE IS TAKEN TO AVOID
C     ARTIFICIALLY STRONG COOLING RATES AT THE FIRST GRID
C     POINT IN THE CLEAR AIR ABOVE CLOUD TOP.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:   ROLF LANGLAND (NOARL)
C                       HOWARD LEWIT (CSC)
C                       LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):
C
C        CALL CORCR(QR,FPO,FMH,TR,X1U,X2U,GAM2,EO,EH,A,PRATIO,QL,
C                   IM1,Z,C1,C2,C3,WK,L2,ENU,LL,FM,FP,FN)
C
C INPUT PARAMETERS:
C
C   FPO    = THE UPWARD RADIATIVE FLUX FROM THE SURFACE
C   FMH    = THE DOWNWARD FLUX AT THE "TOP" OF THE ATMOSPHERE
C   TR     = THE TRANSMISSION FUNCTION COMPUTED IN CORWL/CORWS
C   X1U    = THE "UPPER ATMOSPHERE" WATER VAPOR PATH INTEGRAL
C   X2U    = THE ATMOSPHERIC DENSITY INTEGRAL
C   GAM2   = THE SURFACE ALBEDO
C   EO     = THE UPWARD EMITTED FLUX FROM THE SURFACE
C   EH     = THE DOWNWARD EMITTED FLUX FROM THE TOP OF ATMOS.
C   A      = AN ABSORPTION COEFFICIENT FOR CLOUD DROPLETS
C   PRATIO = THE EXNER FUNCTION
C   QL     = THE LIQUID WATER CONTENT
C   IM1    = NN - 1
C   Z      = ARRAY OF GRID POINT HEIGHTS
C   C1,C2,C3 = ARRAYS FOR COMPUTING VERTICAL DERIVATIVES
C   WK     = GENERAL PURPOSE WORK ARRAY
C   L2     = INDEX DETERMINES WHETHER WK(I,3) OR WK(I,4) USED
C   ENU    = UPPER ATMOSPHERE BLACK BODY FUNCTION VALUES
C   LL     = NUMBER OF "UPPER ATMOSPHERE" POINTS --SEE S/R INITIAL
C
C
```

```
C   OUTPUT PARAMETERS:
C
C   QR    = THE COMPUTED RADIATIVE HEATING/ COOLING RATES
C   FM    = DOWNWARD RADIATIVE FLUX
C   FP    = UPWARD RADIATIVE FLUX
C   FN    = NET FLUX AT EACH GRID LEVEL (=FP(I)-FM(I))
C
C......................MAINTENANCE SECTION..........................
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C......................END PROLOGUE.................................




      SUBROUTINE CORRD
C
C........................START PROLOGUE.............................
C
C SUBPROGRAM NAME:   CORRD
C
C DESCRIPTION:
C
C     THIS IS THE MAIN RADIATION SUBPROGRAM.  FIRST INTEGRATED
C     COLUMN ABUNDANCES OF WATER VAPOR AND CARBON DIOXIDE ARE
C     COMPUTED, ALONG WITH THE BLACK BODY FUNCTION AT EACH LEVEL.
C     THE SOLAR FLUX AT THE TOP OF THE ATMOSPHERE IS DEFINED
C     BASED ON THE LOCAL HOUR ANGLE AND SOLAR DECLINATION.  THEN
C     CALLS TO S/R CORCR ARE MADE: FIRST TO GET THE LONGWAVE
C     FLUX AND HEATING RATES, THEN TO GET THE SHORTWAVE VALUES.
C     MUCH OF THIS CODING IS BASED ON A.R.A.P. REPORT NO. 289,
C     1976.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:    ROLF LANGLAND (NOARL)
C                        HOWARD LEWIT (CSC)
C                        LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):  CALL CORRD
C
C........................MAINTENANCE SECTION........................
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C........................END PROLOGUE...............................
```

```
      SUBROUTINE CORWL(XL1,XL2,XL3,AL,WK,N)
C
C.........................START PROLOGUE.............................
C
C SUBPROGRAM NAME:   CORWL
C
C DESCRIPTION:
C
C     THIS SUBPROGRAM COMPUTES THE TRANSM'SSION
C     FUNCTIONS FOR CALCULATING RADIATIVE FLUXES.
C     CORWL IS FOR LONGWAVE TRANSMISSION FUNCTIONS, WHILE
C     CORWS IS FOR THE SHORTWAVE TRANSMISSION FUNCTIONS.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:   ROLF LANGLAND (NOARL)
C                       HOWARD LEWIT (CSC)
C                       LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):  CALL CORWL(XL1,XL2,XL3,AL,WK,N)
C
C     INPUT PARAMETERS:
C
C   XL1  = DENSITY PATH INTEGRAL OF WATER VAPOR FOR TRANSMISSION FCTN.
C   XL2  = DENSITY INTEGRAL OF CO2 (LONGWAVE) OR AEROSOL (SHORTWAVE)
C   XL2  = LIQUID WATER DENSITY INTEGRAL
C   AL   = CLOUD WATER ABSORPTION COEFFICIENT
C   N    = NUMBER OF GRID POINTS
C
C     OUTPUT PARAMETERS:
C
C   WK   = LONGWAVE TRANSMISSION FUNCTION
C
C.........................MAINTENANCE SECTION........................
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C.........................END PROLOGUE...............................
```

```
      SUBROUTINE CORWS (XL1,XL2,XL3,AL,WK,N)
C
C.........................START PROLOGUE.............................
C
C SUBPROGRAM NAME:   CORWS
C
C DESCRIPTION:
C
C       THIS SUBPROGRAM COMPUTES THE TRANSMISSION
C       FUNCTIONS FOR CALCULATING RADIATIVE FLUXES.
C       CORWL IS FOR LONGWAVE TRANSMISSION FUNCTIONS, WHILE
C       CORWS IS FOR THE SHORTWAVE TRANSMISSION FUNCTIONS.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:   ROLF LANGLAND (NOARL)
C                       HOWARD LEWIT (CSC)
C                       LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):  CALL CORWS(XL1,XL2,XL3,AL,WK,N)
C
C       INPUT PARAMETERS:
C
C    XL1  = DENSITY PATH INTEGRAL OF WATER VAPOR FOR TRANSMISSION FCTN.
C    XL2  = DENSITY INTEGRAL OF CO2 (LONGWAVE) OR AEROSOL (SHORTWAVE)
C    XL2  = LIQUID WATER DENSITY INTEGRAL
C    AL   = CLOUD WATER ABSORPTION COEFFICIENT
C    N    = NUMBER OF GRID POINTS
C
C       OUTPUT PARAMETERS:
C
C    WK   = SHORTWAVE TRANSMISSION FUNCTION
C
C.........................MAINTENANCE SECTION........................
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C.........................END PROLOGUE...............................
```

```
      SUBROUTINE ESAT(ES1,T1,N)
C
C.........................START PROLOGUE..............................
C
C SUBPROGRAM NAME:    ESAT
C
C DESCRIPTION:
C
C     THIS SUBPROGRAM COMPUTES THE SATURATION VAPOR PRESSURE (MB)
C     FOR A GIVEN TEMPERATURE (KELVIN) BASED ON LOWE (JAM, 16,
C     100-103, 1977).  FOR TEMPERATURE LESS THAN 223K, CONSTANTS
C     ARE USED BASED ON NEPRF REPORT BY LOWE.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:   ROLF LANGLAND (NOARL)
C                       HOWARD LEWIT (CSC)
C                       LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):  CALL ESAT(ES1,T1)
C
C     INPUT PARAMETERS:
C
C   T1   = TEMPERATURE (KELVIN)
C   N    = NUMBER OF GRID POINTS
C
C     OUTPUT PARAMETERS:
C
C   ES1 = SATURATION VAPOR PRESSURE (MB)
C
C.........................MAINTENANCE SECTION.........................
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C.........................END SECTION.................................
```

```
      SUBROUTINE HOSKEEP
C.........................START PROLOGUE.............................
C
C SUBPROGRAM NAME:   HOSKEEP
CC
C DESCRIPTION:
C
C     THIS SUBPROGRAM COMPUTES THE GRID VOLUME LIQUID WATER
C     CONTENT, QL, AND THE FRACTIONAL CLOUDINESS, RC.  ALSO,
C     THE EDDY TRANSFER COEFFICIENTS ARE COMPUTED --- EITHER
C     USING YAMADA'S OR THE HASSID AND GALPERIN (1983) METHOD.
C     AT LAND POINTS, TIME-DEPENDENT BUDGET EQUATIONS FOR SURFACE
C     TEMPERATURE AND MOISTURE ARE SOLVED USING THE FORCE-
C     RESTORE METHOD.  SOMMERIA AND DEARDORFF (1977), HASSID
C     AND GALPERIN (1983), AND DEARDORFF (1978) ARE PRIMARY
C     SOURCES.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:   ROLF LANGLAND (NOARL)
C                       HOWARD LEWIT (CSC)
C                       LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):  CALL HOSKEEP
C
C.........................MAINTENANCE SECTION........................
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C.........................END PROLOGUE...............................
```

```
      SUBROUTINE INITIAL
C
C.....................START PROLOGUE.................................
C
C SUBPROGRAM NAME:   INITIAL
C
C DESCRIPTION:
C
C    THIS IS THE INITIALIZATION SUBROUTINE IN WHICH THE NOGAPS
C    OR NORAPS PROFILES ARE READ IN AND INTERPOLATED TO THE NABL
C    GRID.  MANY OF THE MODEL ARRAYS FOR TURBULENCE VARIABLES
C    ARE INITIALIZED AS ZERO HERE.  ALSO, INITIAL VALUES FOR
C    RADIATIVE FLUX CALCULATIONS ABOVE THE TOP OF THE
C    COMPUTATIONAL GRID ARE SET HERE.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:   ROLF LANGLAND (NOARL)
C                       HOWARD LEWIT (CSC)
C                       LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):  CALL INITIAL
C
C.....................MAINTENANCE SECTION............................
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C.....................END PROLOGUE...................................


      PARAMETER (NN=40,NR=21,NL=11,NT=37,NP=20)

      CHARACTER CMOD*6

      DIMENSION RHOAU(NN),RHOVU(NN)

      COMMON/BLK1/ALF,VISK,A1,A2,B1,B2,C,G,BETA,VK,SQ,Z(NN),
     1  BT,R,GAMM,GAMH,HEATL,CPD,RD,GRD,EFLUX(NN),SFLUX(NN)

      COMMON/BLK2/PLOC(10),TTI(NR),QQI(NR),UUI(NR),VVI(NR),PPI(NR),
     1            KBLKST,KTOT,CMOD

      COMMON/BLANK/ABT(NN),ABTH(NN),C1(NN),PS,ENUL(NN),LL,
     2  C2(NN),C3(NN),CLAM2(NN),UN(NN),VN(NN),UF(NN),VF(NN),
     3  DQWDZ(NN),DT,DTHLDZ(NN),DUDZ(NN),DUMV(NN),DVDZ(NN),EKH(NN),
     4  EKM(NN),IM1,IM2,IM3,IT,ITOT,P(NN),X1U(NN),X2U(NN),DTHVDZ(NN),
     5  PRATIO(NN),QL(NN),QRN(NN),QSL(NN),QS(NN),X2UL(NN),
     6  QW(NN),QWSB(NN),QWSTAR,QWZ,RC(NN),RLAT,RLON,IBLKST(NP),
     7  SPECHUM(NN),PHIM,COSALP,SINALP,SST,WK(NN,NL),
     8  TH(NN),THL(NN),THLQWB(NN),THLSB(NN),HDUCT,GAM2,
     9  THLSTAR,THLWB(NN),THLZ,THVWB(NN),TKE(NN),TSIG(NR),
     A  TKESR(NN),TL(NN),TV(NN),U(NN),UG(NN),USTAR,V(NN),VG(NN),
     B  WQWB(NN),XL(NN),THTEND(NN),QTEND(NN),VT(NN),TDEW(NN),
```

```
     C  Y1,Z1,ZL,ZN(NN),ZS(NN),DECLN,ZTIME,UTEND(NN),VTEND(NN),PSTEND,
     D  QRL(NN),QRS(NN),TIMER,IDELAY,XMZ,XBC,ZSIG(NR),XM(NN),
     E  TT1(NL),QQ1(NL),FML(NN),FPL(NN),TSTEND,GW,GWWG,TOPO,
     F  TT2(NL),QQ2(NL),FMS(NN),FPS(NN),
     G  SIGDT1(NL),SIGDT2(NL)
C
C THIS S/R READS THE INITIAL CONDITIONS COMING FROM NORAPS OR NOGAPS.
C THESE INITIAL CONDITIONS AT GRIDPOINT JL ARE:
C
C  PLOC(1)  = IPT
C  PLOC(2)  = JPT
C  PLOC(3)  = LATITUDE
C  PLOC(4)  = LONGITUDE
C  PLOC(5)  = ISEA  (CURRENTLY NOT USED)
C  PLOC(6)  = TSEA
C  PLOC(7)  = PS2
C  PLOC(8)  = GRDWET
C  PLOC(9)  = TOPO
C  PLOC(10) = TSINIT:  NORAPS DEEP SOIL TEMP (CURRENTLY NOT OUTPUT
C                                         BY OPERATIONAL NORAPS)
C  TTI(K)   = TEMP AT SIGMA LEVEL
C       (HERE K=1 EQUALS K=KK IN NORAPS, ETC.)
C  QQI(K)   = SPECIFIC HUMIDITY
C  UUI(K)   = U-COMPONENT WIND
C  VVI(K)   = V-COMPONENT WIND
C  PPI(K)   = INITIAL PRESSURE ON SIGMA LEVELS
C
```

```
      SUBROUTINE NABL42(JTAU,NTAU)
C
C.......................START PROLOGUE...........................
C
C SUBPROGAM NAME:  NABL42
C
C DESCRIPTION:  WRITE NABL FORECAST TO FILE (FNOC FORMAT) FOR SKEW-T
C
C ORIGINAL PROGRAMMER: ROLF LANGLAND (1991)
C
C CURRENT PROGRAMMER:  ROLF LANGLAND (NOARL)
C                      LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):  CALL NABL42(JTAU,NTAU)
C       SET NABL NAMELIST VARIABLE "FNSKEW" = .TRUE.
C
C INPUT PARAMETERS:
C       JTAU  - CORRESPONDS TO ITAU(M) NABL FORECAST TAU
C       NTAU  - YYMMDDHH OF NABL FORECAST TAU (CH*8)
C
C OUTPUT PARAMETERS:
C      RLAT   - LATITUDE (0-90), INS (N OR S)
C      BLON   - LONGITUDE (0-180), IEW (E OR W)
C      CMOD   - NORAPS OR NOGAPS IDENTIFIER
C      P      - PRESSURE (MB)
C      T99    - TEMPERATURE (C)
C      TD99   - DEW POINT TEMPERATURE (C)
C      IV9D   - WIND DIRECTION IND DEG (0-360)
C      IV99   - WIND SPEED (KNOTS)
C      ZFT    - NABL GRID POINT HEIGHT, INC TERRAIN (FT)
C      TOPO   - TERRAIN HEIGHT (FT)
C
C OUTPUT FILES:  TAPE42
C
C.......................MAINTENENCE SECTION......................
C
C RECORD OF CHANGES:
C
C.......................END PROLOGUE.............................
```

```
      SUBROUTINE NABL46(JTAU,NTAU)
C
C........................START PROLOGUE............................
C
C SUBPROGRAM NAME:  NABL46
C
C DESCRIPTION:  WRITE REFRACTIVITY DATA TO FILE IN FNOC FORMAT
C
C ORIGINAL PROGRAMMER:  ROLF LANGLAND
C
C CURRENT PROGRAMMER:   ROLF LANGLAND
C                       LIANA ZAMBRESKY
C
C USAGE (CALLING SEQUENCE):  CALL NABL46(JTAU,NTAU)
C       SET NABL NAMELIST VARIABLE "FNRFRC" = .TRUE.
C
C INPUT PARAMETERS:
C       JTAU  - CORRESPONDS TO ITAU(M) NABL FORECAST TAU
C       NTAU  - YYMMDDHH OF NABL FORECAST TAU (CH*8)
C
C OUTPUT PARAMETERS:
C      RLAT   - LATITUDE (0-90), INS (N OR S)
C      BLON   - LONGITUDE (0-180), IEW (E OR W)
C      CMOD   - NORAPS OR NOGAPS IDENTIFIER
C      P      - PRESSURE (MB)
C      T99    - TEMPERATURE (C)
C      TD99   - DEW POINT TEMPERATURE (C)
C      XMZ    - SURFACE MODIFIED REFRACTIVITY
C      TOPO   - TERRAIN HEIGHT (FT)
C      XM     - MODIFIED REFRACTIVITY
C      ZMDEL  - LAPSE RATE OF REFRACTIVITY
C      RFRAC  - REFRACTIVITY CATEGORY
C      ZFT    - NABL GRID POINT HEIGHT, INC TERRAIN (FT)
C
C OUTPUT FILES:  TAPE46
C
C........................MAINTENENCE SECTION.......................
C
C RECORD OF CHANGES:
C
C........................END PROLOGUE..............................
```

```
      SUBROUTINE NABL75(ITIME,NDTG,IMOD)
C
C........................START PROLOGUE............................
C
C SUBPROGRAM NAME:  NABL75
C
C DESCRIPTION:  WRITE NABL FORECAST TO FILE FOR SKEW-T PLOT
C
C ORIGINAL PROGRAMMER:  ROLF LANGLAND (1991)
C
C CURRENT PROGRAMMER:   ROLF LANGLAND (NOARL)
C                       LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):  CALL NABL75(ITIME,NDTG,IMOD)
C       SET NABL NAMELIST VARIABLE "NBSKEW" = .TRUE.
C
C INPUT PARAMETERS:
C       ITIME - NABL FORECAST TAU (INTEGER)
C       NDTG  - YYMMDDHH OF NABL FCST TAU (I8)
C       IMOD  - 1=NABL/NORAPS, 2=NABL/NOGAPS
C
C OUTPUT PARAMETERS:
C       KBLKST - BLOCKSTATION IDENTIFIER (5-DIGIT) FOR FORECAST POINT
C       PLOC(1)- NORAPS/NOGAPS GRID POINT IN EAST-WEST DIRECTION
C       PLOC(2)- NORAPS/NOGAPS GRID POINT IN NORTH-SOUTH DIRECTION
C       PLOC(5)- NORAPS/NOGAPS LAND-SEA-ICE CODE
C       PLOC(9)- TERRAIN HEIGHT (M)
C       RLAT - LATITUDE (-90 S-HEM TO +90 N-HEM) OF FORECAST POINT
C       DLON - LONGITUDE (-180 E-HEM TO +180 W-HEM) OF FORECAST POINT
C       GWWG   - GROUND MOISTURE PARAMETER
C       SST    - SEA SURFACE TEMPERATURE, OR GROUND SKIN TEMPERATURE
C       P    - PRESSURE (MB)
C       T99  - TEMPERATURE (C)
C       TD99 - DEW POINT TEMPERATURE (C)
C       UV9D - WIND DIRECTION IN DEG (0-360)
C       UV99 - WIND SPEED (M/S)
C       IH   - NABL GRID POINT HEIGHT, INC TERRAIN (M)
C
C OUTPUT FILES:  TAPE75
C
C........................MAINTENENCE SECTION.......................
C
C RECORD OF CHANGES
C
C........................END PROLOGUE..............................
```

```
      SUBROUTINE NABL80(ITIME,NDTG,JL,NK)
C
C........................START PROLOGUE............................
C
C SUBPROGRAM NAME:  NABL80
C
C DESCRIPTION:  WRITE NORAPS FORECAST TO FILE FOR SKEW-T PLOT
C
C ORIGINAL PROGRAMMER:  ROLF LANGLAND (1991)
C
C CURRENT PROGRAMMER:  ROLF LANGLAND (NOARL)
C                      LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):  CALL NABL80(ITIME,NDTG,JL,NK)
C       SET NABL NAMELIST VARIABLE "NRSKEW" = .TRUE.
C
C INPUT PARAMETERS:
C       ITIME - NABL FORECAST TAU (INTEGER)
C       NDTG  - YYMMDDHH OF NABL FORECAST TAU (I8)
C       JL    - COUNTER FOR SEQUENCE OF NABL FORECAST POINTS
C       NK    - COUNTER FOR SEQUENCE OF NORAPS SKEW-T SAVE TIMES
C
C OUTPUT PARAMETERS:
C       KBLKST - BLOCKSTATION IDENTIFIER (5-DIGIT) FOR FORECAST POINT
C       RLAT   - LATITUDE (-90 S-HEM TO +90 N-HEM) OF FORECAST POINT
C       DLON   - LONGITUDE (-180 E-HEM TO +180 W-HEM) OF FORECAST POINT
C       IMOD   - ( = 4) IDENTIFIES NORAPS IN SKEW-T SOFTWARE
C       SLOC3(1,.) - NORAPS GRID POINT IN X-DIRECTION
C       SLOC3(2,.) - NORAPS GRID POINT IN Y-DIRECTION
C       PLOC(8)- GROUND MOISTURE PARAMETER
C       PLOC(9)- TERRAIN HEIGHT (M)
C       PLOC(5)- NORAPS LAND-SEA-ICE CODE
C       SSTNR  - SEA SURFACE TEMPERATURE OR GROUND SKIN TEMPERATURE
C       PPS     - PRESSURE (MB)
C       T99     - TEMPERATURE (C)
C       TD99    - DEW POINT TEMPERATURE (C)
C       UV9D    - WIND DIRECTION IN DEG (0-360)
C       UV99    - WIND SPEED (M/S)
C       NRZ     - HEIGHT OF NORAPS SIGMA LEVELS (M)
C
C OUTPUT FILES:  TAPE80
C
C........................MAINTENENCE SECTION.......................
C
C RECORD OF CHANGES:
C
C........................END PROLOGUE..............................
```

```
      SUBROUTINE OUT(IL,JL,CLAT,CLON,DSKIP)
C
C.......................START PROLOGUE...............................
C
C SUBPROGRAM NAME:        OUT
C
C DESCRIPTION:
C
C     THIS IS THE MAIN OUTPUT SUBROUTINE FOR NABL. LOGICAL
C     VARIABLE, DSKIP, IS USED TO CONTROL THE VOLUME OF
C     OUTPUT ONE GETS.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:    ROLF LANGLAND (NOARL)
C                        HOWARD LEWIT (CSC)
C                        LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):  CALL OUT(IL,JL,CLAT,CLON,DSKIP)
C
C INPUT PARAMETERS:
C
C     IL,JL     = NOGAPS/NORAPS I,J VALUES FOR SELECTED NABL POINT
C     CLAT,CLON = NOGAPS/NORAPS LATITUDE & LONGITUDE OF NABL POINT
C     DSKIP     = LOGICAL VARIABLE USED TO DETERMINE WHETHER TO SKIP
C                 PRINTING OUT A BLOCK OF INFORMATION
C
C.......................MAINTENANCE SECTION..........................
C
C PRINIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C.......................END PROLOGUE.................................
```

```
      SUBROUTINE TMDAG(NDTG,MT)
C
C SUBPROGRAM NAME:  TMDAG
C
C.......................START PROLOGUE...........................
C
C DESCRIPTION:  WRITE SELECTED NABL FORECAST VARIABLES TO OUTPUT
C               FILE FOR TIME SERIES PLOTTING
C
C ORIGINAL PROGRAMMER:  ROLF LANGLAND
C
C CURRENT PROGRAMMER:   ROLF LANGLAND
C                       LIANA ZAMBRESKY
C
C USAGE (CALLING SEQUENCE):  CALL TMDAG(NDTG,MT)
C       SET NABL NAMELIST VARIABLE "TMSER" = .TRUE.
C
C INPUT PARAMETERS:
C       NDTG  - YYMMDDHH OF NABL FORECAST TAU (18)
C       MT    - INTEGER FLAG FOR TAU-0 OR NOT TAU-0
C
C OUTPUT PARAMETERS:
C      KBLKST - BLOCKSTATION IDENTIFIER (5-DIGIT) OF FORECAST POINT
C      THL    - LIQUID WATER POTENTIAL TEMPERATURE (K)
C      QW     - TOTAL WATER CONTENT (KG/KG)
C      UV99   - WIND SPEED (M/S)
C      UV9D   - WIND DIRECTION (0-360 DEG)
C      TKE    - TURBULENT KINETIC ENERGY
C      THVWB  - TURBULENT SENSIBLE HEAT FLUX (WATTS PER SQUARE METER)
C      WQWB   - TURBULENT LATENT HEAT FLUX (WATTS PER SQUARE METER)
C
C OUTPUT FILES:  TAPE70 (ALIAS PDATA)
C
C.......................MAINTENENCE SECTION......................
C
C RECORD OF CHANGES:
C
C.......................END PROLOGUE.............................
```

```
      SUBROUTINE TQTEND(IHR)
C
C........................START PROLOGUE..............................
C
C SUBPROGRAM NAME:   TQTEND
C
C DESCRIPTION:
C
C     THIS SUBPROGRAM COMPUTES THE HOURLY AVERAGE ADIABATIC
C     TENDENCIES FOR TEMPERATURE AND MOISTURE BASED ON NOGAPS/
C     NORAPS OUTPUT.  CODE WITHIN NOGAPS/NORAPS SAVES THE
C     TEMPERATURES AND MOISTURES THAT RESULT AT HOURLY INTERVALS
C     WHEN ONLY ADIABATIC (ADVECTIVE) PROCESSES ARE CONSIDERED.
C     THUS, THESE TENDENCIES, ONCE INTERPOLATED TO THE NABL
C     GRID, BECOME THE ADIABATIC, ADVECTIVE FORCING THAT GIVES
C     NABL ITS 3-D CHARACTER --, I.E., THEY PERMIT NABL TO
C     RESPOND TO SYNOPTIC CHANGES AND SUBSIDENCE IN A MANNER
C     THAT A STRICTLY 1-D MODEL WOULD NOT.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:   ROLF LANGLAND (NOARL)
C                       HOWARD LEWIT (CSC)
C                       LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE): CALL TQTEND(IHR)
C
C INPUT PARAMETERS:
C
C   IHR = FORECAST HOUR AT WHICH THIS SUBROUTINE IS CALLED
C
C........................MAINTENANCE SECTION.........................
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C........................END PROLOGUE................................
```

```
      SUBROUTINE UTENDS(IHR)
C
C.......................START PROLOGUE..............................
C
C SUBPROGRAM NAME:     UTENDS
C
C DESCRIPTION:
C
C    THIS SUBPROGRAM COMPUTES THE WIND TENDENCIES BASED ON
C    THE NOGAPS/NORAPS WIND COMPONENTS AT ONE HOUR INTERVALS.
C    THESE TENDENCIES ARE USED TO NUDGE THE NABL WIND FORECAST
C    TOWARDS THE NOGAPS/NORAPS FORECAST SO THAT IN THE ABSENCE
C    OF TURBULENT MIXING THE NABL FORECAST WINDS ARE VERY
C    SIMILAR TO THE NOGAPS/NORAPS WINDS.  THE TENDENCIES ARE
C    ALSO INTERPOLATED TO THE NABL GRID HERE.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:   ROLF LANGLAND (NOARL)
C                       HOWARD LEWIT (CSC)
C                       LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE): CALL UTENDS(IHR)
C
C      INPUT PARAMETERS:
C
C   IHR  = FORECAST HOUR AT WHICH THIS SUBROUTINE IS CALLED
C
C.......................MAINTENANCE SECTION.........................
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C.......................END PROLOGUE................................
C
```

```
      SUBROUTINE UVCOMP(PHI,PHIN,EKM,ZS,ZN,WK,DT,USTAR,ANG,Z)

C...........................START PROLOGUE...........................
C
C SUBPROGRAM NAME:    UVCOMP
C
C DESCRIPTION:
C
C     THIS SUBPROGRAM COMPUTES THE MOMENTUM FLUX DIVERGENCE
C     FOR A GIVEN WIND COMPONENT.  A TRIDIAGONAL IMPLICIT
C     EQUATION IS SOLVED FOR THIS EDDY MIXING PROCESS.  THE
C     SOLUTION IS NUDGED TOWARDS THE NOGAPS/NORAPS WINDS ---
C     THIS NUDGING BEING DONE TOWARDS THE END OF THE MAIN
C     PROGRAM.  IN REGIONS WHERE THE MIXING IS WEAK, THE
C     NUDGING WILL CAUSE THE NABL WIND PROFILE TO EVOLVE
C     SIMILAR TO NOGAPS/NORAPS.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:   ROLF LANGLAND (NOARL)
C                       HOWARD LEWIT (CSC)
C                       LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):
C
C        CALL UVCOMP(PHI,PHIN,EKM,ZS,ZN,WK,DT,USTAR,ANG,Z)
C
C INPUT PARAMETERS:
C
C   PHI   = WIND COMPONENT (U OR V) AT TIME LEVEL N
C   EKM   = EDDY COEFFICIENT FOR MOMENTUM
C   ZS    = GRID SPACING DIFFERENCE BETWEEN CURRENT LEVEL, I, AND I-1.
C   ZN    = GRID SPACING DIFFERENCE BETWEEN CURRENT LEVEL, I, AND I+1.
C   WK    = GENERAL PURPOSE WORK ARRAY
C   DT    = TIME STEP (SEC)
C   USTAR = FRICTION VELOCITY ( M/S )
C   ANG   = COSINE OR SINE OF SURFACE WIND TO POSITIVE X-AXIS
C   Z     = GRID POINT HEIGHT ARRAY
C
C      OUTPUT PARAMETERS:
C
C   PHIN  = VALUE OF WIND COMPONENT (U OR V) AT NEW TIME LEVEL, N+1.
C
C...........................MAINTENANCE SECTION.....................
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C...........................END PROLOGUE............................
```

```
      SUBROUTINE UVSD (U,V,SPD,DIR,N)
C
C...........................START PROLOGUE..........................
C
C SUBPROGRAM NAME:  UVSD
C
C DESCRIPTION:
C
C     THIS SUBPROGRAM TAKES U AND V WIND COMPONENTS AND CONVERTS
C     THEM TO SPEED (M/S) AND DIRECTION (DEG).
C
C ORIGINAL PROGRAMMER, DATE:   ROLF LANGLAND (NOARL), 1990
C
C CURRENT PROGRAMMER:    ROLF LANGLAND (NOARL)
C                        HOWARD LEWIT (CSC)
C                        LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE): CALL UVSD(U,V,SPD,DIR)
C
C...........................MAINTENANCE SECTION.....................
C
C     INPUT PARAMETERS:
C
C   U = EAST-WEST WIND COMPONENT ( M/S )
C   V = NORTH-SOUTH WIND COMPONENT ( M/S )
C
C     OUTPUT PARAMETERS:
C
C   SPD = TOTAL WIND SPEED ( M/S )
C   DIR = WIND DIRECTION FROM NORTH ( DEG )
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C...........................END PROLOGUE............................
```

```
      SUBROUTINE VINTRP(X,Z,XSD,J1,K1,IEXTRP)
C
C.......................START PROLOGUE..............................
C
C SUBPROGRAM NAME:    VINTRP
C
C DESCRIPTION:
C
C    THIS SUBPROGRAM INTERPOLATES NOGAPS OR NORAPS VALUES TO
C    THE NABL GRID.  THE INTERPOLATION IS DONE LINEARLY IN
C    HEIGHT.  IF IEXTRP=1, NABL POINTS BELOW THE LOWEST NOGAPS/
C    NORAPS POINT ARE SET EQUAL TO THE NOGAPS/NORAPS VALUE AT
C    THEIR LOWEST GRID POINT ABOVE THE SURFACE.  IF IEXTRP IS
C    NOT EQUAL TO ONE, THEN NABL POINTS BELOW THE LOWEST NOGAPS/
C    NORAPS POINT ARE FOUND BY EXTRAPOLATING THE NEAR SURFACE
C    NOGAPS/NORAPS GRADIENT DOWNWARD.
C
C ORIGINAL PROGRAMMER, DATE:   STEPHEN BURK (NOARL), 1987
C
C CURRENT PROGRAMMER:   ROLF LANGLAND (NOARL)
C                       HOWARD LEWIT (CSC)
C                       LIANA ZAMBRESKY (FNOC)
C
C USAGE (CALLING SEQUENCE):  CALL VINTRP(X,Z,XSD,J1,K1,IEXTRP)
C
C      INPUT PARAMETERS:
C
C   Z      = NABL GRID POINT HEIGHTS
C   XSD    = VARIABLE VALUE AT NOGAPS/NORAPS LEVEL
C   J1,K1  = VALUES USED TO SET SECOND INDEX OF ARRAY XSD
C   IEXTRP = DETERMINES NATURE OF INTERPOLATION/ EXTRAPOLATION
C            FOR NABL POINTS LOWER THAN LOWEST NOGAPS/NORAPS PT.
C
C      OUTPUT PARAMETERS:
C
C   X    = INTERPOLATED VALUE ON NABL GRID OF THE INPUT VARIABLE, XSD.
C
C.......................MAINTENANCE SECTION.........................
C
C PRINCIPLE VARIABLES AND ARRAYS:
C
C RECORD OF CHANGES:
C
C.......................END PROLOGUE................................
```

```
      PROGRAM NGNPNT(INPUT,OUTPUT,TER,TAPE1=INPUT,TAPE2=TER,
     * PRO,TAPE7=PRO,TAPE8)
C
C..................START PROLOGUE.....................................
C
C  PROGRAM NAME: PROGRAM NGNPNT
C
C  DESCRIPTION:
C  INTERACTIVE ENTRY OF NOGAPS NABL POINTS FOR TRANSFER TO
C   DGFILE ON SAM
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT,  20 DEC 1990
C
C  CURRENT PROGRAMMER:
C
C  CONTRACT NUMBER AND TITLE: NONE
C
C  REFERENCES: NONE
C
C  COMPUTER/OPERATING SYSTEM: FNOC NOS BE PEPS
C
C  LIBRARIES OF RESIDENCE:OPSPL1
C
C  CLASSIFICATION: UNCLAS
C
C  USAGE (JCL):
C  EXECUTED IN INTERACTIVE PROCEDURE NGNABL:
C  APLIB(MT1731,*NGNPNT)
C NGNPNT.
C (SEE PROCEDURE NGPOINT,OPSFIL FOR TREATMENT OF OUTPUT)
C
C  INPUT FILES:  TAPE1= INTERACTIVE USER INPUT
C
C  OUTPUT FILES:
C  TAPE2 = DIRECTIVES TO INTERACTIVE USER
C  TAPE8 = DATA FOR TRANSFER TO SAM
C
C  ERROR CONDITIONS:
C  NUMEROUS MESSAGES TO USER FOR INCORRECT ENTRIES AND FORMATS
C
C  ADDITIONAL COMMENTS:
C
C..................MAINTENANCE SECTION................................
C
C  BRIEF DESCRIPTION OF PROGRAM MODULES:
C  SUBROUTINE NGOLD ( & ENTRY NGWRT) READ AND WRITE THE
C     EXISTING POINT FROM FILE NGNREC
C  SUBROUTINE NGRPT - USER INPUT NGPOINT LOCATION
C  SUBROUTINE NGFORM - CHANGES THE POINTS TO SAM INPUT FORMAT FOR
C                TAPE 8
C
C  DATA FILES: FILE NGNREC CONTAINS POINTS ON FRONT-END FILE
C           FOR REFERENCE AND BACKUP
C
C  TEMPORARY FILES: NONE
```

```
C
C  PRINCIPAL VARIABLES AND ARRAYS: SEE COMDECK DISCRIPTIONS
C
C  COMMON BLOCKS: SEE COMDECKS
C
C  METHOD:
C  DISPLAYS EXISTING POINTS TO USER, PROMPTS CHANGES TO INDIVIDUAL
C  POINTS. USER ENTERS POINTS IN STANDARD FORMAT , PROGRAM CHANGES
C   THEM TO NOGAPS REQUIRED FORMAT AND WRITES TO TAPE8
C   PROCEDURE NGNMOD TRANSFERS TAPE8 TO SAM AND WRITES TO
C   DGFILE.
C
C  LANGUAGE: FTN5
C
C  RECORD OF CHANGES:
C <<CHANGE NOTICE>> NGNPNT*01 (9 JAN 1991) -- LEWIT
C   MODIFY COMMENTS
C..................END PROLOGUE......................................
```

```
      SUBROUTINE NGEXTR(LINE,IS,IN,IT,FF,NUM,RR,NX,*)
C...................START PROLOGUE.....................................
C
C  SUBPROGRAM NAME: SUBROUTINE NGEXTR
C
C  DESCRIPTION:
C   EXTRACT CHARACTERS, INTEGERS OR REAL NUMBERS FROM CHARACTER
C    STRING
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C  CALL NGEXTR(LINE,IS,IN,IT,FF,NUM,RR,NX,*1)
C   ALTERNATE RETURN *1 FOR BLANK INPUT
C
C  INPUT PARAMETERS:
C  LINE - 80 CHARACTER VARIABLE
C  IS - STARTING POINT IN LINE
C   IN - MAXIMUM NUMBER OF CHARACTERS TO SEARCH
C  IT - OUTPUT TYPE
C      0 - CHARACTER
C      1 - INTEGER
C      2 - REAL NUMBER
C
C  OUTPUT PARAMETERS:
C  FF - CHARACTER STRING
C  NUM - INTEGER
C  RR - REAL NUMBER
C
C  INPUT FILES: NONE
C
C  OUTPUT FILES: NONE
C
C  COMMON BLOCKS: NONE
C
C  ERROR CONDITIONS: NONE
C
C...................MAINTENANCE SECTION................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C  SEE INPUT/OUTPUT PARAMETERS
C   FIRST - FLAG FOR FIRST UNIT
C   NCHR - NUMBER OF UNIT COUNT
C   LDEC - LOCATION IN NCHR OF DECIMAL TYPE 2
C   KNA/KNB - ASCII NUMBER RANGE
C   KCA/KCB  - ASCII LETTER RANGE
C  NSGN - (-) SIGN FLAG
```

```
C
C  METHOD:
C  IDENTIFY EACH CHARACTER BY TYPE AND ACCUMULATE THEM
C   ACCORDING TO DECODE TYPE.
C   PROVIDE (-) IF NOT CHARACTER AND (.) IF TYPE 2
C
C  RECORD OF CHANGES:
C <<CHANGE NOTICE>> NGEXTR*01 (20 MAR 91) -- LEWIT
C  FIX DECIMAL PLACE FOR REAL NUMBERS
C
C...................END PROLOGUE......................................
```

```
      SUBROUTINE NGFORM
C
C...................START PROLOGUE......................................
C
C  SUBPROGRAM NAME: SUBROUTINE NGFORM
C
C  DESCRIPTION:
C  CONVERT TO NOGAPS INPUT FORM AND WRITE OUT IN NAMELIST FORMAT
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C  CALL NGFORM
C
C   INPUT PARAMETERS:
C  ARRAYS OF LAT, LONG AND LAND/OCEAN
C
C  OUTPUT PARAMETERS:
C  NONE
C
C   INPUT FILES: NONE
C
C  OUTPUT FILES: TAPE 8
C
C  COMMON BLOCKS: $NGPNT
C
C  ERROR CONDITIONS: BAD NAMELIST WRITE
C
C...................MAINTENANCE SECTION.................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C  SEE INPUT/OUTPUT PARAMETERS
C
C  METHOD:
C  CONVERT LAT, LONG TO NOGAPS INPUT FORMAT
C  LAT NORTH = +, SOUTH = -
C  LONG E = +, WEST = -
C   WRITE NAMELIST TO TAPE 8 FOR TRANSFER TO SAM
C
C  RECORD OF CHANGES:
C
C <<CHANGE NOTICE>> NGFORM*01 (20 MAR 1991) -- LEWIT
C  INCREASE LAT/LON UP TO 3 DECIMAL PLACES
C...................END PROLOGUE........................................
```

```
      SUBROUTINE NGLL(AL,NS,AN,EW,O)
C
C...................START PROLOGUE....................................
C
C  SUBPROGRAM NAME: SUBROUTINE NGLL
C
C  DESCRIPTION:
C   PROVIDE FOR INPUT OF LAT/LONG POSITIONS
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C   CALL NGLL(AL,NS,AN,EW,O)
C
C  INPUT PARAMETERS:
C   NONE
C
C  OUTPUT PARAMETERS:
C  AL,NS  = LAT
C  AN,EW  = LONG
C  O = LAND/OCEAN
C
C  INPUT FILES:TAPE1
C
C  OUTPUT FILES: TAPE2
C
C  COMMON BLOCKS:
C
C  ERROR CONDITIONS:
C  NUMEROUS MESSAGES TO USERS IN CASE OF INVALID INPUT
C    UP TO NG ERRORS ALLOWED THEN MESSAGE TO GET HELP AND STOP
C
C...................MAINTENANCE SECTION................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C   LINE- 80 CHARACTER USER INPUT
C   ITRY - ERROR COUNT
C
C  METHOD:
C  PROMPT USER TO ENTER FULL LAT LONG. USER SUBROUTINE NGEXTR TO
C    EXTRACT THE NUMBERS AND CHARACTERS FROM THE ENTRY. CHECK THE
C     ENTRY FOR VALID VALUES.  VERIFY WITH USER.
C     IF NOT CORRECT TRY AGAIN NG TIMES.
C    USER ENTER "LAND" OR "OCEAN"
C
C  RECORD OF CHANGES:
C <<CHANGE NOTICE>> NGLL*01 (20 MAR 91) -- LEWIT
```

```
C  INCREASE LAT/LON UP TO 3 DECIMAL PLACES
C
C...................END PROLOGUE.....................................
```

```
      SUBROUTINE NGOLD
C
C...................START PROLOGUE......................................
C
C  SUBPROGRAM NAME: SUBROUTINE NGOLD
C
C  DESCRIPTION:
C  READ AND WRITE POINT INFORMATION IN STANDARD FORMAT AS
C  PERMANENT FILE NGNREC, ID=OP, SN=SHARDSK
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C  CALL NGOLD
C
C  INPUT PARAMETERS:
C  ARRAYS OF LAT, LONG AND LAND/OCEAN
C
C  OUTPUT PARAMETERS:
C  NONE
C
C  INPUT FILES: NONE
C
C  OUTPUT FILES: ZRANDIO RECORD NGNREC
C  OUTPUT FILE: PERMANENT FILE NGNREC ID=OP , SN=SHARDSK
C  COMMON BLOCKS: $NGPNT
C
C  ERROR CONDITIONS: RECORD NOT AVAILABLE CONTINUE GOING
C
C...................MAINTENANCE SECTION.................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C  SEE INPUT/OUTPUT PARAMETERS
C
C  METHOD:
C  READ AND WRITE THE DATA AS UNFORMATTED RECORDS TO
C    TAPE6
C  CATALOG NEW CYCLE OF NGNREC
C   PURGE OLDEST CYCLE OF NGNREC
C
C  RECORD OF CHANGES:
C
C <<CHANGE NOTICE>> NGOLD*01 (9 JAN 1991) -- LEWIT
C   MODIFY COMMENTS
C <<CHANGE NOTICE>> NGOLD*02 (24 APRIL 1991) -- LEWIT
C  LIMIT NUMBER OF POINTS READ IN TO NPT
C...................END PROLOGUE........................................
```

```
      SUBROUTINE NGRPT
C
C...................START PROLOGUE......................................
C
C  SUBPROGRAM NAME: SUBROUTINE NGRPT
C
C  DESCRIPTION:
C  USER CHANGE, DELETE OR ADD POINTS TO DGFILE
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C  CALL NGRPT
C
C   INPUT PARAMETERS: NONE
C
C  OUTPUT PARAMETERS: LAT AND LONG IN ARRAYS ALAT AND ALON
C
C   INPUT FILES: NONE
C
C  OUTPUT FILES: TAPE2
C
C  COMMON BLOCKS: NGPNT
C
C  ERROR CONDITIONS:
C  LIMIT OF NPT ENTRY POINTS, MESSAGE AND CONTINUE
C   INCORRECT FORMAT OF POINT
C
C...................MAINTENANCE SECTION.................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C   AL - USER  INPUT LAT
C   AN - USER INPUT LONG
C   NS/EW - USER INPUT N OR S AND E OR W
C   NMP - COUNT OF NUMBER OF INPUT POINTS
C   ALAT - LAT ARRAY
C   ALON - LONG ARRAY
C   LS = LAND/OCEAN ARRAY
C
C  METHOD:
C  GET USER INPUT OF LAT, LONG AND LAND/SEA VIA SUBROUTINE NGLL
C  SAVE MODIFIED OR ADDED DATA
C   TEST FOR ADDITIONAL INPUT POINTS UP TO NPT
C  DELETE INDICATED POINT, CLOSE SPACES IN ARRAY
C
C  RECORD OF CHANGES:
C
```

```
C <<CHANGE NOTICE>> NGRPT*01 (9 JAN 1991) -- LEWIT
C   MODIFY COMMENTS, CORRECT INITIAL DATA LIST DISPLAY
C<<CHANGE NOTICE>> NGRPT*02 (20 MAR 1991)-- LEWIT
C  INCREASE LAT/LON UP TO 3 DECIMAL PLACES
C..................END PROLOGUE....................................


      SUBROUTINE NGSTOP(IP)
C
C..................START PROLOGUE...................................
C
C  SUBPROGRAM NAME: NGSTOP
C
C  DESCRIPTION:
C  TERMINATE THE PROGRAM
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C   CALL NGSTOP(IP)
C
C  INPUT PARAMETERS:
C   IP - FLAG 1= NORMAL, 2 = ABNORMAL
C
C  OUTPUT PARAMETERS: NONE
C
C  INPUT FILES: NONE
C
C  OUTPUT FILES: NONE
C
C  COMMON BLOCKS: NONE
C
C  ERROR CONDITIONS: NONE
C
C..................MAINTENANCE SECTION..............................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C  IP - FLAG
C
C  METHOD:
C  STOP PROGRAM WITH MESSAGE NORMAL OR ABNORMAL TERMINATION
C    DISCONNECT TAPE1( INPUT) AND TAPE2( OUTPUT)
C      STOP
C
C  RECORD OF CHANGES:
C
C..................END PROLOGUE....................................
```

```
      SUBROUTINE NGYN(*,*)
C
C...................START PROLOGUE.....................................
C
C  SUBPROGRAM NAME: SUBROUTINE NGYN
C
C  DESCRIPTION:
C  PROVIDE PROMPT TO USER TO ENTER Y OR N
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C  CALL NGYN(*1,*2)
C   ALTERNATE RETURNS:
C     *1  Y RESPONSE
C     *2  N RESPONSE
C
C  INPUT PARAMETERS: NONE
C
C  OUTPUT PARAMETERS: NONE
C
C  INPUT FILES: TAPE1
C
C  OUTPUT FILES: TAPE2
C
C  COMMON BLOCKS: NONE
C
C  ERROR CONDITIONS:
C   USER MUST ANSWER Y OR N, KEEP TRYING
C
C...................MAINTENANCE SECTION................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C   YN - USER RESPONSE
C
C  METHOD:
C  PROMPT USER TO ENTER Y OR N. IF INCORRECT KEEP TRYING
C
C  RECORD OF CHANGES:
C
C...................END PROLOGUE.......................................
```

```
      PROGRAM NRPPNT(INPUT,OUTPUT,TER,TAPE1=INPUT,TAPE2=TER,
     * PRO,TAPE7=PRO,TAPE8)
C
C....................START PROLOGUE......................................
C
C  PROGRAM NAME: PROGRAM NRPPNT
C
C  DESCRIPTION:
C  INTERACTIVE ENTRY OF NORAPS NABL POINTS FOR TRANSFER TO
C   NRNFIL7 ON SAM
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT,  20 DEC 1990
C
C  CURRENT PROGRAMMER:
C
C  CONTRACT NUMBER AND TITLE: NONE
C
C  REFERENCES: NONE
C
C  COMPUTER/OPERATING SYSTEM: FNOC NOS BE PEPS
C
C  LIBRARIES OF RESIDENCE:OPSPL1
C
C  CLASSIFICATION: UNCLAS
C
C  USAGE (JCL):
C  EXECUTED IN INTERACTIVE PROCEDURE NRPNABL:
C  APLIB(MT1731,*NRPPNT)
C NRPPNT.
C (SEE PROCEDURE NRPNABL,OPSFIL FOR TREATMENT OF OUTPUT)
C
C  INPUT FILES:  TAPE1= INTERACTIVE USER INPUT
C
C  OUTPUT FILES:
C  TAPE2 = DIRECTIVES TO INTERACTIVE USER
C  TAPE8 = DATA FOR TRANSFER TO SAM
C
C  ERROR CONDITIONS:
C  NUMEROUS MESSAGES TO USER FOR INCORRECT ENTRIES AND FORMATS
C
C  ADDITIONAL COMMENTS:
C   NONE
C....................MAINTENANCE SECTION.................................
C
C  BRIEF DESCRIPTION OF PROGRAM MODULES:
C  SUBROUTINE NROLD - READ AND WRITE THE EXISTING POINTS FROM PF
C                     NRNREC(AREA)
C  SUBROUTINE NRRPT - USER INPUT NRPNABL LOCATION
C  SUBROUTINE NRFORM - CHANGES THE POINTS TO SAM INPUT FORMAT FOR
C              TAPE 8
C  SUBROUTINE NRPROC - PREPARE PROCEDURE PRO TO PASS AREA NUMBER TO
C                   PROCEDURE NRNMOD.
C
C  DATA FILES: FILE NRPNABL  CONTAINS FRONT END POINTS FOR
C    REFERENCE
```

```
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS: SEE COMDECK DISCRIPTIONS
C
C  COMMON BLOCKS: SEE COMDECKS
C
C  METHOD:
C  DISPLAYS EXISTING POINTS TO USER, PROMPTS CHANGES TO INDIVIDUAL
C  POINTS. USER ENTERS POINTS IN STANDARD FORMAT , PROGRAM CHANGES
C   THEM TO NORAPS REQUIRED FORMAT AND WRITES TO TAPE8
C   PROCEDURE NRNMOD TRANSFERS TAPE8 TO SAM AND WRITES TO
C   NRNFIL7.
C
C  LANGUAGE: FTN5
C
C  RECORD OF CHANGES:
C..................END PROLOGUE........................................
```

```
      SUBROUTINE NREXTR(LINE,IS,IN,IT,FF,NUM,RR,NX,*)
C
C...................START PROLOGUE.......................................
C
C  SUBPROGRAM NAME: SUBROUTINE NREXTR
C
C  DESCRIPTION:
C   EXTRACT CHARACTERS, INTEGERS OR REAL NUMBERS FROM CHARACTER
C    STRING
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C  CALL NREXTR(LINE,IS,IN,IT,FF,NUM,RR,NX,*1)
C   ALTERNATE RETURN *1 FOR BLANK INPUT
C
C  INPUT PARAMETERS:
C  LINE - 80 CHARACTER VARIABLE
C  IS - STARTING POINT IN LINE
C   IN - MAXIMUM NUMBER OF CHARACTERS TO SEARCH
C  IT - OUTPUT TYPE
C      0 - CHARACTER
C      1 - INTEGER
C      2 - REAL NUMBER
C
C  OUTPUT PARAMETERS:
C  FF - CHARACTER STRING
C  NUM - INTEGER
C  RR - REAL NUMBER
C
C  INPUT FILES: NONE
C
C  OUTPUT FILES: NONE
C
C  COMMON BLOCKS: NONE
C
C  ERROR CONDITIONS: NONE
C
C...................MAINTENANCE SECTION..................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C  SEE INPUT/OUTPUT PARAMETERS
C   FIRST - FLAG FOR FIRST UNIT
C   NCHR - NUMBER OF UNIT COUNT
C   IDEC - LOCATION IN NCHR OF DECIMAL TYPE 2
C   KNA/KNB - ASCII NUMBER RANGE
C   KCA/KCB  - ASCII LETTER RANGE
```

```
C  NSGN - (-) SIGN FLAG
C
C  METHOD:
C   IDENTIFY EACH CHARACTER BY TYPE AND ACCUMULATE THEM
C    ACCORDING TO DECODE TYPE.
C    PROVIDE (-) IF NOT CHARACTER AND (.) IF TYPE 2
C
C  RECORD OF CHANGES:
C
C...................END PROLOGUE.......................................
```

```
      SUBROUTINE NRFORM
C
C...................START PROLOGUE......................................
C
C  SUBPROGRAM NAME: SUBROUTINE NRFORM
C
C  DESCRIPTION:
C  CONVERT TO NORAPS INPUT FORM AND WRITE OUT IN NORAPS INPUT FORMAT
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C  CALL NRFORM
C
C   INPUT PARAMETERS:
C  ARRAYS OF LAT, LONG
C
C  OUTPUT PARAMETERS:
C  NONE
C
C   INPUT FILES: NONE
C
C  OUTPUT FILES: TAPE 8
C
C  COMMON BLOCKS: $NRPNT, $NRAREA
C
C  ERROR CONDITIONS: BAD NORAPS INPUT WRITE
C
C...................MAINTENANCE SECTION................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C  SEE INPUT/OUTPUT PARAMETERS
C  MX = MAXIMUM NUMBER OF NORAPS AREAS
C *************
C    ONLY SET UP FOR AREA 7 12/90
C **************
C
C  METHOD:
C  CONVERT LAT, LONG TO NORAPS INPUT FORMAT
C  LAT NORTH = +, SOUTH = -
C  LONG E = +, WEST = -
    WRITE NORAPS INPUT TO TAPE 8 FOR TRANSFER TO SAM

   RECORD OF CHANGES:
    CHANGE NOTICE>> NRFORM*01 (4 SEPT 91) -- LEWIT
       [illegible] 8 DATA
```

```
C...................END PROLOGUE.........................................
C  MAKE TAPE13 INPUT FOR NORAPS. INCLUDES THE LAT/LONG OF THE NABL
C  POINTS AS WELL AS OTHER NORAPS CONSTANTS.
C  CAN BE VARIED BY NORAPS AREA UP TO 9 AREAS.
C
C   EAST LONGITUDE = 360 - LONG
```

```
      SUBROUTINE NRLL(AL,NS,AN,EW)
C
C..................START PROLOGUE.......................................
C
C  SUBPROGRAM NAME: SUBROUTINE NRLL
C
C  DESCRIPTION:
C   PROVIDE FOR INPUT OF LAT/LONG POSITIONS
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C   CALL NRLL(AL,NS,AN,EW)
C
C   INPUT PARAMETERS:
C   NONE
C
C  OUTPUT PARAMETERS:
C  AL,NS  = LAT
C  AN,EW  = LONG
C
C   INPUT FILES:TAPE1
C
C  OUTPUT FILES: TAPE2
C
C  COMMON BLOCKS:
C
C  ERROR CONDITIONS:
C  NUMEROUS MESSAGES TO USERS IN CASE OF INVALID INPUT
C    UP TO NG ERRORS ALLOWED THEN MESSAGE TO GET HELP AND STOP
C
C..................MAINTENANCE SECTION..................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C   LINE- 80 CHARACTER USER INPUT
C   ITRY - ERROR COUNT
C
C  METHOD:
C  PROMPT USER TO ENTER FULL LAT LONG. USER SUBROUTINE NREXTR TO
C    EXTRACT THE NUMBERS AND CHARACTERS FROM THE ENTRY. CHECK THE
C     ENTRY FOR VALID VALUES.  VERIFY WITH USER.
C     IF NOT CORRECT TRY AGAIN NG TIMES.
C
C  RECORD OF CHANGES:
C
C..................END PROLOGUE.........................................
```

```
      SUBROUTINE NROLD
C
C..................START PROLOGUE......................................
C
C  SUBPROGRAM NAME: SUBROUTINE NROLD
C
C  DESCRIPTION:
C  READ AND WRITE POINT INFORMATION IN STANDARD FORMAT AS
C  PERMANENT FILE NRNREC(AREA), ID=OP, SN=SHARDSK
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C  CALL NROLD
C
C  INPUT PARAMETERS:
C  ARRAYS OF LAT, LONG
C
C  OUTPUT PARAMETERS:
C  NONE
C
C  INPUT FILES: NONE
C
C  OUTPUT FILES: PERMANENT FILE RECORD NRNREC(AREA)
C
C  COMMON BLOCKS: $NRPNT
C
C  ERROR CONDITIONS: RECORD NOT AVAILABLE CONTINUE GOING
C
C..................MAINTENANCE SECTION.................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C  SEE INPUT/OUTPUT PARAMETERS
C
C  METHOD:
C  HAVE THE USER INPUT THE AREA TO BE UPDATED. IF AREA IS NOT DEFINED
C    FOR NABL, NOTIFY THE USER AND ABORT.
C  READ AND WRITE ALL VARIABLES AS UNFORMATTED.
C   NON-CONVERTED REAL NUMBERS OR AS INTEGER REPRESENTATIONS
C    OF CHARACTERS. WRITE TO RECORD NRNREC(AREA) ON FILE ID=OP.
C   REVERSE THE PROCEDURE WHEN READING.
C
C  RECORD OF CHANGES:
C <<CHANGE NOTICE>> NROLD*C1 (4 SEPT 91) -- LEWIT
C  ADD AREA 8
C
C..................END PROLOGUE........................................
```

```
      SUBROUTINE NRPROC
C
C...................START PROLOGUE.....................................
C
C  SUBPROGRAM NAME: SUBROUTINE NRPROC
C
C  DESCRIPTION:
C  PREPARE PROCEDURE TO PASS AREA  NUMBER TO NRNMOD
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C  CALL NRPROC
C
C  INPUT PARAMETERS:
C  NONE
C
C  OUTPUT PARAMETERS: NONE
C
C  INPUT FILES: NONE
C
C  OUTPUT FILES:
C   TAPE7 ( PRO)
C
C  COMMON BLOCKS:
C   $NRAREA
C
C  ERROR CONDITIONS: NONE
C
C...................MAINTENANCE SECTION................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C  NAREA =  NORAPS AREA NUMBER
C
C  METHOD:
C  SET UP PROCEDURE WITH AREA NUMBER INSERTED
C
C  RECORD OF CHANGES:
C
C...................END PROLOGUE.......................................
```

```
      SUBROUTINE NRRPT
C
C..................START PROLOGUE.....................................
C
C  SUBPROGRAM NAME: SUBROUTINE NRRPT
C
C  DESCRIPTION:
C  USER CHANGE, DELETE OR ADD POINTS TO NRNFIL7
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C  CALL NRRPT
C
C   INPUT PARAMETERS: NONE
C
C  OUTPUT PARAMETERS: LAT AND LONG IN ARRAYS ALAT AND ALON
C
C   INPUT FILES: NONE
C
C  OUTPUT FILES: TAPE2
C
C  COMMON BLOCKS:$NRPNT
C
C  ERROR CONDITIONS:
C  LIMIT OF NPT ENTRY POINTS, MESSAGE AND CONTINUE
C   INCORRECT FORMAT OF POINT
C
C..................MAINTENANCE SECTION................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C   AL - USER  INPUT LAT
C   AN - USER INPUT LONG
C   NS/EW - USER INPUT N OR S AND E OR W
C   NMP - COUNT OF NUMBER OF INPUT POINTS
C   ALAT - LAT ARRAY
C   ALON - LONG ARRAY
C
C  METHOD:
C  GET USER INPUT OF LAT, LONG   VIA SUBROUTINE NRLL
C  SAVE MODIFIED OR ADDED DATA
C   TEST FOR ADDITIONAL INPUT POINTS UP TO NPT
C   DELETE INDICATED POINT, CLOSE SPACE IN ARRAYL
C
C  RECORD OF CHANGES:
C
C..................END PROLOGUE.......................................
```

```
      SUBROUTINE NRSTOP(IP)
C
C...................START PROLOGUE.......................................
C
C  SUBPROGRAM NAME: NRSTOP
C
C  DESCRIPTION:
C  TERMINATE THE PROGRAM
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C   CALL NRSTOP(IP)
C
C   INPUT PARAMETERS:
C    IP - FLAG 1= NORMAL, 2 = ABNORMAL
C
C  OUTPUT PARAMETERS: NONE
C
C   INPUT FILES: NONE
C
C  OUTPUT FILES: NONE
C
C  COMMON BLOCKS: NONE
C
C  ERROR CONDITIONS: NONE
C
C...................MAINTENANCE SECTION..................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C   IP - FLAG
C
C  METHOD:
C  STOP PROGRAM WITH MESSAGE NORMAL OR ABNORMAL TERMINATION
C    DISCONNECT TAPE1( INPUT) AND TAPE2( OUTPUT)
C     STOP
C
C  RECORD OF CHANGES:
C
C...................END PROLOGUE.........................................
```

```
      SUBROUTINE NRYN(*,*)
C
C...................START PROLOGUE.......................................
C
C  SUBPROGRAM NAME: SUBROUTINE NRYN
C
C  DESCRIPTION:
C  PROVIDE PROMPT TO USER TO ENTER Y OR N
C
C  ORIGINAL PROGRAMMER, DATE: HOWARD LEWIT, DECEMBER 1990
C
C  CONTRACT NUMBER AND TITLE:NONE
C
C  CLASSIFICATION: UNCLASS
C
C  USAGE (CALLING SEQUENCE):
C  CALL NRYN(*1,*2)
C   ALTERNATE RETURNS:
C      *1   Y RESPONSE
C      *2   N RESPONSE
C
C  INPUT PARAMETERS: NONE
C
C  OUTPUT PARAMETERS: NONE
C
C  INPUT FILES: TAPE1
C
C  OUTPUT FILES: TAPE2
C
C  COMMON BLOCKS: NONE
C
C  ERROR CONDITIONS:
C   USER MUST ANSWER Y OR N, KEEP TRYING
C
C...................MAINTENANCE SECTION..................................
C
C  DATA FILES: NONE
C
C  TEMPORARY FILES: NONE
C
C  PRINCIPAL VARIABLES AND ARRAYS:
C   YN - USER RESPONSE
C
C  METHOD:
C  PROMPT USER TO ENTER Y OR N. IF INCORRECT KEEP TRYING
C
C  RECORD OF CHANGES:
C
C...................END PROLOGUE.........................................
```

Appendix D. Data Element Lists

USER_ONLINE_TERMINAL Data Elements

Element Name	Description	Type/ Units	Limits/Range
AREA	Numeric identifier of larger-scale model.	numeric	as displayed by prompt
ACTION	Basic action - add, delete or display a point.	char*1	"Y" or "N" response to prompts
POINT_NR	Sequence number of point concerned.	numeric	as displayed by prompt
LAT_DEG	Absolute latitude of data point.	real deg.	0.0 to 90.0
LAT_HEMI	Hemisphere of lat.	char*1	"N" or "S"
LON_DEG	Absolute longitude data point.	real deg.	0.0 to 180.0
LON_HEMI	Hemisphere of long.	char*1	"E" or "W"
LANDORSEA	Does user consider point "wet" or "dry".	char*5	"LAND" or "OCEAN"

Note: the above item is not currently implemented for NORAPS.

MAX_TAU	Maximum tau of data to be written out.	numeric	0 to L-scale model's limit.

Note: the above item is not currently implemented.

DATAPOINT_DETAILS/DATAPOINT_NAMELIST Data Elements

Element Name	Description	Type/ Units	Limits/Range
CAREA	Identifier of larger-scale forecast model.	char*10	ex."NORAPS7"
MAXTAU	Maximum hourly tau to be output by the model.	integer	0 to L-scale model's limit
NRPTS	Number of points in file.	integer	1 - 20 (NABL dimen'n limit)
PTLAT	Latitude of a point.	real	+/-0.0 - 90.0
PTLON	West longitude of point.	real	+0.0 - 360.0
PLAND	Land or Ocean indicator.	char*5	"LAND" or "OCEAN"

Note: the three variables above will be repeated NRPTS times in each model-unique file.

NAMELIST_DATA Data Elements

Element Name	Description	Type/ Units	Limits/Range
NLTAUF	Final (highest) tau of NABL forecast.	integer (HH)	0(dflt) - 36 (dimensioned upper limit)
NTAUF	Number output taus (number of entries in ITAU below).	integer	0(dflt) - 5 (dimensioned upper limit)
NFCST	Number of locations (number of IPT/JPT or ALAT/ALON points).	integer	0(dflt) - 20 (dimensioned upper limit)
ITYPE	Type of location points provided (e.g. I/J or LAT/LON).	char*2	"IJ"(dflt) or "LL"
IPT	Comma delimited list of 1 to 20 I coordinates (on NOGAPS or NORAPS grid) where forecasts are desired.	int(20)	larger-scale model determines
JPT	- - - J coordinate equivalent of IPT above - - -		
Note:	The IJ capability provided by the three variables above is really DIAGNOSTIC and not required operationally.		
ITAU	Comma delimited list of taus for the NABL forecast output.	int(5)	00 - NLTAUF
ALAT	Comma delimited list of 1 to 20 latitudes where forecasts are desired.	real(20)	+/-0.0 to 90.0
ALON	- West longitude match for ALAT above -		+0.0 to 360.0
NBLKST	Comma delimited list of 1 to 20 (WMO) block-station idents corresponding to IPT/JPT or ALAT/LON pairs. -OPTIONAL-	integer (BKSTA)	five digits
NRSKEW	Indicates when SKEWT type output from the large-scale model input is desired. This is DIAGNOSTIC INFORMATION which is written to TAPE80.	logical	.TRUE. or .FALSE.(dflt)
NRTAU	Comma delimited list of 1 to 8 times at which NRSKEW output (see above) desired. No meaning if NRSKEW is FALSE.	int(8)	00 to NLTAUF

- this table is continued on next page -

Element Name	Description	Type/ Units	Limits/Range
NSKEW	Number of entries in NRTAU above. No meaning if NRSKEW is FALSE.	integer	0(dflt) - 8 (dimensioned upper limit)
DSKIP	Indicates when limited data is to be written to OUTPUT. Limits the DIAGNOSTIC information when set.	logical	.TRUE.(dflt) or .FALSE.
TSER	Indicates when DIAGNOSTIC time series information is to be written to TAPE70. Increases DIAGNOSTIC information when set.	logical	.TRUE. or .FALSE.(dflt)
NBSKEW	Indicates when the optional SKEWT data in NOARL (metric units) format is to be written to TAPE75.	logical	.TRUE. or .FALSE.(dflt)
FNSKEW	Indicates when standard SKEWT data in FNOC (english units) format is to be written to TAPE42.	logical	.TRUE. or .FALSE.(dflt)
FNREFC	Indicates when standard refractive effects data is to be written to TAPE46.	logical	.TRUE. or .FALSE.(dflt)

Note: PARAMETER statements are correctly used to define some key array dimensions and loop control variables used in the NABL model. For example, the upper dimensions of NLTAUF and NFCST above are controlled by PARAMETER statements within the appropriate program modules.

INITIAL_COND_DATA Data Elements

Element Name	Description	Type/ Units	Limits/Range
CMOD	Ident of larger-scale forecast model.	char*6	"NOGAPS" or "NORAPS"
CLBL	Data-set label. (UNUSED)	char*20	any string
IDIM	Number of geographic points in this data set.	integer	1 - 20 (dimensioned limit)
IGMT	GMT hour of starting time of the forecast (tau=0).	integer	0 or 12
DECL	Declination of sun at time IFTIM (below).	real deg.	+/-00.0 to +/-23.5
IFTIM	GMT of tau zero (forecast base-time).	char*8	YYMMDDHH
IFCST	GMT of maximum forecast tau.	char*8	YYMMDDHH
NRTAUF	Max (last) tau in data set.	integer	0 - max fcst (NABL limit 36)
PL_IPT	I coord. of the data point on the forecast grid.	real	grid limits
PL_JPT	J coord. of the data point on the forecast grid.	real	grid limits

Note: For NOGAPS this is the lower-left grid point in the "four grid point set" containing the LAT/LON below. For NORAPS this is the nearest grid point to the LAT/LON below. Also, see Appendix E.

PL_LAT	Latitude of the point.	real deg.	+/-0.0 - 90.0
PL_LON	West longitude of the point.	real deg.	+0.0 - 360.0

Note: This is the user's requested LAT/LON, not the LAT/LON of I/J above (unless they coincidently are the same).

PL_LRS	"Land-or-Sea" flag in F3.1 (x.y) format (see note below)	real	0.0 to 0.3 or 1.0 to 1.3

Note: "x" (to left of decimal) indicates user's request (if any) where 0 is water (the NOGAPS default) and 1 is land (the NORAPS default). "y" (to right of decimal) indicates actual conditions at the I/J grid point - 0 is water, 1 is land, 2 is sea-ice and 3 is land-ice. Also, see discussion in Appendix E.

- this table is continued on the next page -

Element Name	Description	Type/ Units	Limits/Range
PL_TSFC	Surface temperature.	realdeg.K	positive
PL_PSFC	Surface pressure.	real(mb.)	positive
PL_GWET	Ground wetness.	real (ND)	0.0 - 1.0
PL_TRHT	Terrain height.	real (m)	n/a
PL_DST	Deep soil temperature. (Not avail from NOGAPS.)	realdeg.K	positive

Note: For NOGAPS the five variables above and all of the level variables below are conditions interpolated to the requested LAT/LON from the "four grid point set" containing that LAT/LON. For NORAPS they are values extracted from the single grid point nearest to the requested LAT/LON. Also, see discussion in Appendix E.

TTI	Level temperature.	realdeg.K	positive
QQI	Level specific humidity.	real (kg/kg)	positive
UUI	Level X component of wind.	real (mps)	+/-
VVI	Level Y component of wind.	real (mps)	+/-
PPI	Level pressure.	real (mb.)	positive

Note: the five variables above are provided for 18 (NOGAPS) or 21 (NORAPS) forecast levels (from lowest to highest) in the tau zero data sets only. This same tau zero information for the first 11 levels is repeated but in a different sequence following the surface data below. This redundancy may be worthy of elimination at some future time.

PSFC	- same as PL_PSFC above (one could be eliminated) -		
TSFC	- same as PL_TSFC above (one could be eliminated) -		
GWET	- same as PL_GWET above (one could be eliminated) -		
Z1L	Surface roughness height.	real (m)	positive
ID	- same as PL_IPT above (one could be eliminated) -		
JD	- same as PL_JPT above (one could be eliminated) -		

- this table is continued on the next page -

Element Name	Description	Type/ Units	Limits/Range
TTDN	Exact tau of data set which depends on time-step in larger-scale model which might be a few seconds off.	real (ND)	positive
TADB	- same as data element TTI above -		
QQHR	- same as data element QQI above -		
SIGP	- same as data element PPI above -		
UUHR	- same as data element UUI above -		
VVHR	- same as data element VVI above -		

Note: the five variables above are provided for the first 11 model (NOGAPS or NORAPS) forecast levels (from lowest to highest) for all tau data sets including tau zero. See the related note following data element PPI above.

COMPLETE_FORECAST_DATA Data Elements

Element Name	Description	Type/ Units	Limits/Range
CAREA	Identifier of larger-scale forecast model.	char*10	ex."NORAPS7"
NLTAUF	Final (highest) tau of NABL forecast.	integer (HH)	0 - 36 (dimensioned limit)
NTAUF	Nr. of output taus (number of entries in ITAU below).	integer	0(dflt) - 5 (dimen.limit)
ITAU	Comma delimited list of the NTAUF taus in NABL output.	integer	00 - NLTAUF
NRPTS	Number of forecast points.	integer	1 - 20 (dimensioned limit)
ALAT	Comma delimited list of NRPTS latitudes of the points.	real	+/-0.0 to 90.0
ALON	- West longitude match for ALAT above -		+0.0 to 360.0
NRLEVS	Number of levels in the NABL forecast.	integer	40 (until/ unless chg'd.)
LEVHTS	Comma delimited list of NABL level heights (lowest first).	integer (m)	positive
TAUSET	Tau-set identifier (based on on ITAU sequence.)	char*5 (TAUhh)	ex. "TAU06", "TAU36"
PTSET	Point-set identifier (based on NRPTS/ALAT/ALON sequence.	char*5 (PTnnn)	ex. "PT29 ", "PT8", "PT100"
PL_LRS	"Land-or-Sea" flag in F3.1 (x.y) format (see note below)	real	0.0 to 0.3 or 1.0 to 1.3

Note: "x" (to left of decimal) indicates user's request (if any) where 0 is water (the NOGAPS default) and 1 is land (the NORAPS default). "y" (to right of decimal) indicates actual conditions at the I/J grid point - 0 is water, 1 is land, 2 is sea-ice and 3 is land-ice. Also, see discussion in Appendix E.

PL_TRHT	Terrain height.	real (m)	n/a

Note: the three variables just above will be repeated NRPTS times in each model-unique file.

- this table is continued on the next page -

Element Name	Description	Type/ Units	Limits/Range
SFCPRES	Surface pressure.	real(mb.)	positive
GNDWET	Ground wetness.	real (ND)	0.0 - 1.0
USTAR	Surface friction velocity.	real (mps)	positive
THLSTAR	Surface scaling parameter for temperature.	real (K)	+/-
MOOBLEN	Monin-Obukhov length - scale of boundary layer turbulence.	real (m)	+/-
SFCRUFF	Surface roughness length.	real (m)	positive
EVDCTHT	Evaporative duct height.	real (m)	positive
SFCMVAL	Surface M-refractive index.	real (ND)	positive

Note: the eight variables above (starting with SFCPRES on the previous page will be repeated NTAUF times within each point set (PTSET).

GPPRESS	Grid pt. pressure.	real (mb)	positive
DDWND	Grid pt. wind direction.	int(deg)	0 - 360
FFWND	Grid pt. wind velocity.	real(mps)	positive
ABSTEMP	Grid pt. absolute temp.	real (K)	positive
POTTEMP	Grid pt. potential temp.	real (K)	positive
VPOTEMP	Grid pt. virtual potential temperature.	real (K)	positive
DPTEMP	Grid pt. dew point temp.	real (K)	positive
SPECHUM	Grid pt. specific humidity.	real(g/kg)	positive
RELHUM	Grid pt. relative humidity.	int. (%)	0 - 100
MVAL	Grid pt. M-refractive index.	real (ND)	positive
CLOUDS	Grid pt. cloud fraction.	real (ND)	0.0 - 1.0
TURBNRG	Grid pt. turbulent kinetic energy.	real (m^2/sec^2)	positive
LHTFLUX	Grid pt. latent heat flux.	real ($watts/m^2$)	+/-
SHTFLUX	Grid pt. sensible heat flux.	real ($watts/m^2$)	+/-

Note: the last 14 variables on the previous page (starting with GPPRESS) will be repeated NRLEVS times in LEVHTS-order within each point-tau set (TAUSET).

Appendix E. NABL Forecasts in the Coastal Zone

The coastal zone presents particular problems to the NABL forecast system as it is now configured. The near-shore output which NABLS provides may not be as accurate as possible in some cases and may be unnecessarily misleading in others. In this appendix those problems are discussed and some recommended changes to improve the existing situation are presented.

THE PROBLEM. When a user specifies a point (latitude/longitude) near a shoreline at which an NABL forecast is desired, it may be assumed that the user expects the base (the surface point) of that forecast to be specifically over water (the ocean, sea, gulf, bay, estuary or whatever) or specifically over land (somewhere above mean sea level). The user might reasonably expect the system to adjust the point slightly to ensure the expected "wet-" or "dry-based" retrieval; or, if that is not possible, the user may reasonably expect the system to provide some other type of data and to inform the user of that substitution.

As the NABL System is now implemented, it does not always know what the user is expecting; and, in those cases when the system does know, the user may receive something different than what was requested without ever being notified. Or, it is conceivable, that the user could be told that a forecast is based on data over land when in fact the data better represents conditions over water (or vice versa). Such problems relate to:

a. the user interface (the NABL user is not always prompted to specify "land" or "sea").

b. the larger-scale-model interface (the larger scale models do not require such user input).

c. the larger-scale-model's data retrieval conventions (these are inconsistent and somewhat arbitrary as will be detailed below).

d. the NABL output formats (NABLS' failure to properly display for the user what land-sea information is now being provided by the larger scale models).

THE EXISTING SYSTEM. The NABLS as presently implemented prompts the user for a "land" or "ocean" assessment with respect to the specified retrieval point only when preparing NAMELIST input for NOGAPS. This is done in program NGNPNT. No such prompting is done by program NRPPNT when preparing NAMELIST input for NORAPS. Nor, does either program provide for a sophisticated user who may wish to make a "sea-ice-" or "land-ice-based" retrieval - even though both NOGAPS and NORAPS are capable of making such distinctions.

Neither NOGAPS nor NORAPS require a user's assessment as to whether the point is "wet" or "dry" though both are programmed to accept such input. If none is provided, NOGAPS defaults to "ocean" and NORAPS defaults to "land" as the "requested condi-

tion". Both larger scale programs then proceed to ignore both the request and the default.

NOGAPS locates the four-grid-point "square" containing the requested latitude and longitude and interpolates from those four corner grid points to the requested geographic point. It returns the interpolated value and the "base condition" of the lower-left (least I and least J) grid point of the four-grid-point set. This "retrieved condition", as determined by the model, may be "water" (assigned a value of 0), "land" (assigned a value of 1), "sea ice" (assigned a value of 2), or "land ice" (assigned a value of 3). The "requested (or default) condition" and the "retrieved condition" are combined into a type real "land-or-sea-status flag" in x.y (F3.1) format [this is variable PLOC(5) in program NABL's COMMON block /BLANK/] - where x is 0 or 1 depending on whether the user requested (or the system defaulted to) "ocean" or the user requested "land"; and y is 0, 1, 2, or 3 (based on the lower-left corner as specified just above). [Note: if NOGAPS is provided I/J coordinates rather than LAT/LON coordinate (as the NAMELIST input format permits, but the present user interface does not) there is no interpolation. The exact value and condition of the requested I/J, regardless of any user-requested condition, are extracted and returned.]

Thus, for example, a user of NOGAPS data may request "land" data and receive pure "ocean" data if the requested point falls ever so slightly within an all ocean grid point "square". In such a case the returned "land-or-sea-status (x.y) flag" would correctly indicate the discrepancy (would take the value "1.0"). In another case, however, the user might request "ocean" data and receive data interpolated from three "land ice" points and only one "ocean" point. In such a case, and if the "ocean" grid point were the lower-left point, the returned "land-or-sea-status flag" would indicate no such problem (it would take the value "0.0").

NORAPS locates the grid-point nearest to the requested latitude and longitude. It returns the value at that nearest grid point regardless of whether or not it is "wet" or "dry". The "retrieved condition" of that nearest grid point ("land", "ocean", "sea ice" or "land ice") is combined with the "requested (or default) condition" and returned in the same x.y (F3.1) format as used and discussed above for NOGAPS.

Thus, for example, a user of NORAPS data may request "land" data and receive pure "sea ice" data if the requested point fall ever so slightly closer to a "sea ice" point than to a "land" point. In such a case the "land-or-sea-status (x.y) flag" would correctly indicate the discrepancy (would take the value "1.2").

In none of its output does NABLS take note of and highlight for the user any discrepancy between "requested" and "retrieved" conditions. REF_EFFECTS_DATA and SKEWT_DATA outputs in FNOC format both derive their "WATER SURFACE" or "LAND SURFACE" indicators from the ground wetness. This provides a reasonably correct wet or dry data assessment, but fails to alert the user should the base condition for the retrieval be different than what was requested. The SKEWT output in alternate NOARL format prints the real value of the "land-or-sea-status flag" in F3.1 format but does not highlight any differences between the integer and fractional components of that flag.

RECOMMENDED CHANGES. To improve NABL forecasts in the coastal zone the following modifications to NABLS, NOGAPS and NORAPS are recommended.

- Programs NGNPNT and NRPPNT should be modified (preferably combined) to provide a common user interface which always prompts for the user's "base condition" assessment as to whether the requested point location is over "land" or "ocean".

Note: The current version of the NABL program is not designed for and should not be used to prepare forecasts over sea ice or continental ice. However, two additional user-input options for "sea-ice" or "land-ice" could be useful for a sophisticated user who recognizes NABL's limitations but who may still wish to take advantage of the larger-scale models' ability to provide over-ice data. Until such time as the Program NABL code has been changed and certified for higher-latitude over-ice applications, any new user-options to retrieve sea- or land-ice data should be appropriately discussed in the NABL users manual and the reader should be clearly cautioned about their use.

- NOGAPS should be modified to always consider the user's "base condition" assessment and to try to retrieve the appropriate type of information without contamination from other types of "base conditions". Thus the NOGAPS program would only interpolate (or conceivably extrapolate) from points of the user's requested type. For example, if two of the four points in the four-grid-point "square" containing the requested LAT/LON location were of the requested type, data would be retrieved by linear interpolation between those two grid points only. A similar scheme would be used if only three points matched. If only one point matched, that single point's values would be extracted (or perhaps some extrapolation using a second nearby point of the same type could be used). If all four points in the "square" were of an unrequested type, the majority points only (the two, three, or four points of the same unrequested other type) would be used for extraction and the type of that majority would be returned as the fractional component in the "land-or-sea-status flag".

- NORAPS should be changed to always extract values from the nearest grid point to the user's requested location which is of the same type as the user's stated "base condition" - <u>unless</u> none of the four grid points in the four-grid-point "square" containing the requested location are of the requested type. In the exception case just stated, the absolute nearest grid point should be used for data retrieval and that point's type should be returned as the fractional component in the "land-or-sea-status flag".

- For consistency with NOGAPS, the NORAPS "base condition" default should be changed to "ocean" - which is the more prevalent type on a global basis and more apt to be of interest to the Navy user at sea.

- Program NABL should be changed to boldly flag any instance of a larger-scale model returning data based on a grid point(s) of an unrequested type. This might take the form:

* * USER REQUESTED LAND DATA - MODEL RETURNED SEA ICE DATA * *

Such a statement would be derived from the "land-or-sea-status flag" (variable PLOC(5) in program NABL's COMMON block /BLANK/) and could be routinely placed at the bottom of each point-tau set of tabular data. If such a statement were made routine, the existing "LAND SFC" or "WATER SFC" output elements should be replaced with the actual ground wetness values (e.g."GW=1.0000").

DISTRIBUTION

NOARL
ATTN: CODE 104
JCSSC, MS 39529-5004

NOARL
ATTN: CODE 125L (10)
JCSSC, MS 39529-5004

NOARL
ATTN: CODE 125P
JCSSC, MS 39529-5004

COMFLENUMOCEANCEN (6)
ATTN: OPERATIONS OFFICER
MONTEREY, CA 93943-5005

COMNAVOCEANCOM
ATTN: CODE N5
JCSSC, MS 39529-5000

REPORT DOCUMENTATION PAGE

Form Approved OBM No. 0704-0188

1. Agency Use Only *(Leave blank).*

2. Report Date. December 1991

3. Report Type and Dates Covered. Final

4. Title and Subtitle.
Navy Atmospheric Boundary Layer (NABL) Model System: Software/Interface Design Document

5. Funding Numbers.
Program Element No. O&M,N
Project No. FNOC
Task No. --
Accession No. DN658754
Work Unit No. 6.3-22FNOC

6. Author(s).
Pierre A. Petit

7. Performing Organization Name(s) and Address(es).
ST Systems Corporation
P.O. Box 51306, Pacific Grove, CA 93950

8. Performing Organization Report Number.
NOARL Technical Note 204

9. Sponsoring/Monitoring Agency Name(s) and Address(es).
Fleet Numerical Oceanography Center
Monterey, CA 93943-5005

Naval Oceanographic and Atmospheric Research Laboratory, Atmospheric Directorate, Monterey, CA 93943-5006

10. Sponsoring/Monitoring Agency Report Number.
NOARL Technical Note 204

11. Supplementary Notes.
Contract no. N00014-88-D-0333

12a. Distribution/Availability Statement.
Approved for public release; distribution is unlimited.

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*

This document describes the complete design of the Navy Atmospheric Boundary Layer (NABL) Model System in terms of its Computer Software Components (CSCs) and Computer Software Units (CSUs).

14. Subject Terms.
Atmospheric boundary layer
NABL model

15. Number of Pages. 107

16. Price Code.

17. Security Classification of Report. UNCLASSIFIED

18. Security Classification of This Page. UNCLASSIFIED

19. Security Classification of Abstract. UNCLASSIFIED

20. Limitation of Abstract. Same as report